A Simulation Study on Manufacturing System Reconfiguration

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy

by Shuqing Xu

December 2008

Abstract

Reconfiguration of manufacturing systems has become a topic of great interest over the past decade. The entire domain of modelling and analysing the reconfiguration of manufacturing systems and machines is developing and expanding but there are still vast areas remain unexplored. The objective of this research is to provide a fundamental insight into how manufacturing systems should be reconfigured in order to cope with the changing market demand. To achieve this objective, a theoretical approach is developed and integrated into a simulation-based model which simulates a medium size manufacturing system.

The research approach consists of three steps using a simulation model. The first step is defining and quantifying machine reconfiguration options through which a manufacturing system is able to cope with the new demand. The second step searches the best reconfiguration options using optimisation algorithms especially the simulated annealing algorithm. The last step comprises the cost and objective functions for measuring the system performance and the efficiency of the reconfigurations. Based upon the assumptions adopted during the research, the simulation experimental results suggest that reconfigurations are essential for the certain manufacturing environments where product life cycles are short yet product demand is variable. Therefore, the simulation and optimisation approach gives a practical way to obtain important information to facilitate manufacturing system reconfiguration.

Acknowledgement

I dedicate this thesis to my parents and thank them for their support and patience throughout this research. I would not have been able work on my thesis without their moral and financial support to fund this research. Their helping hand would always reach out to help me during the most difficult times.

I also want to thank my supervisor Professor K.K.B Hon for his substantial amount of effort. I am grateful for his understanding and patience and for the support that he has offered throughout these years. I would like to thank all my colleagues for their valuable comments and feedback. Last and the most important, many thanks to Robert Armstrong for his help and support.

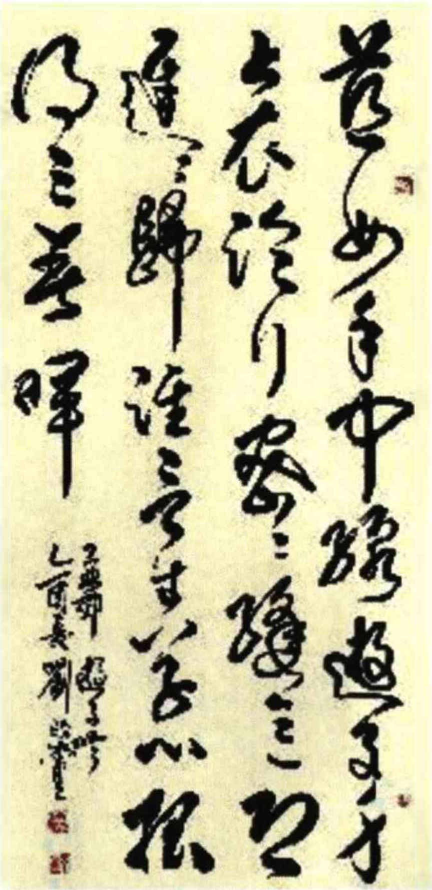


Table of Content

Chapter 7 Product Portfolio Restructuring for the System Reconfiguration

Chapter 8 Conclusion and Future Research

Glossary and Abbreviations

Ant Colony (AC) was inspired by the behaviour of real ants, while almost blind, are capable of finding the shortest path from food sources to the nest. The process is characterized by a positive feedback loop, where the probability increases with the number of previous steps that chose the same path.

Combinatorial Optimisation (CO) deals with discrete problems where the goal is to find the best possible, feasible and discrete solution.

Dedicated manufacturing line (DML) is a machining system designed for production of a specific part type at high volume.

Discrete Event Simulation (DES) model is defined as one in which the state variables change only at those discrete points in time at which events occur.

Flexible Manufacturing System (FMS) is an integrated system of machine modules and material handling equipment under computer control for the automatic random processing of palletized parts.

Genetic Algorithm (GA) is a general-purpose stochastic and parallel search method based on the mechanism of natural selection and natural genetics.

Hill Climbing (HC) is an optimisation technique which belongs to the family of local search. It is a popular first choice as it is a relatively simple technique to implement.

Metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions.

NP problem – "non-deterministically polynomial" – is one that, in the worst case, requires time poly-nominal in the length of the input for solution by a non-deterministic algorithm.

Reconfigurable Machine Tools (RMT) were invented and patented in 1999 in the Engineering Research Centre for Reconfigurable Manufacturing System at the University of Michigan.

Reconfigurable Manufacturing System (RMS) is one designed at the outset for rapid change in its structure, as well as its hardware and software components, in order to quickly adjust its production capacity and functionality within a part family in response to sudden market changes or changes in regulatory requirements.

Scheduling is defined as the process of optimizing resource allocation decisions beforehand.

Simulated Annealing (SA) is a generic probabilistic meta-algorithm for the global optimisation problem, namely locating a good approximation to the global optimum of a given function in a large search space.

Simulation is the imitation of the operation of a real-world process or system over time. Simulation involves the generation of an artificial history of the system, and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system that is represented.

Tabu search (TS) is based on the hill-climbing method that evaluates iteratively a best solution each time the neighbourhood is updated.

Theory of Constraints (TOC) is a business philosophy which seeks to strive towards a global objective, or goal of a system through an understanding of the underlying cause and effect dependency and variation of the system in question.

Throughput is defined as the number of orders the manufacturing system finishes per time unit.

Work-in-process (WIP) is defined as the number of orders in the system which are not finished yet.

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Overview

In modern manufacturing, multiple objectives such as maximum output, minimum operation cost, minimum work in progress, minimum inventory size and maximum customer satisfaction have to be considered or achieved simultaneously. With the liberalization of global markets, customer satisfaction criteria are on a higher level. Competitive pressure has amplified the needs for more agile and flexible processes in which to exploit more opportunities and satisfy more customer requirements. The shortening and dynamics of product life cycles constrain the time actually available for product development and system reconfiguration. Rapid and dynamic product innovation cycles and ever-changing product life cycles are challenging manufacturers to look for more flexible manufacturing processes and production management methods that allow rapid response to market demand within acceptable cost parameters.

Many traditional decisions-making are based on the expertise that exists in each technical field and determined by the personal experience accrued by staffs, ranging from managers to product designers and manufacturing personnel. Although conventional methods and heuristics are available to support decision-making that occurs at each stage of the product life cycle, many of the decisions are based on personal and empirical knowledge. However, due to the large amount of intricately related information and profound complexity, it is not clear, even to the most qualified manufacturing engineer or manager, what may be the best option to achieve particular objectives. With the innovation of computer technology and the application of simulation, the effect of each decision is observed and analysed before its physical implementation. If the outcome is not satisfactory, no cost penalties will be incurred. Therefore simulation is becoming a significant competitive decision making tool due to its ability to predict the effects of changes or to forecast the behavior of a proposed system.

With the ever increasing popularity of simulation, there have been many simulation languages designed and made available to enable an easy entry into the simulation world. This in turn fuels the increase of the popularity of simulation and the field of simulation applications. There are two major types of simulation: continuous and discrete-event though modern simulation language can handle both. Most manufacturing processes fall in the category of discrete-event simulation as the operation of a manufacturing system is represented as a chronological sequence of events. Most modern simulation languages have graphical interfaces and user friendly control systems which are another major step of lowering the entry threshold for simulation applications. A series of comprehensive statistics analysis tools on simulation results are also built into modern languages for gathering more detailed system information. Among many simulation languages, WITNESS by the Lanner Group has been proved to be one of the leading software used for manufacturing simulation and has numerous successful applications in industry (www.lanner.com).

Modern manufacturing world is characterised by ever-increasing demands for process flexibility. Moreover, the uncertainty and variability may reduce the predictability and controllability levels in the system. These complexity dimensions have to be managed in the system design and operational stages. The reconfiguration of manufacturing system is the necessary step for many manufacturing companies. Reconfigurable manufacturing paradigm is a manufacturing philosophy that deals with reconfiguration of manufacturing systems and ensures that manufacturing systems meet current and future demands under turbulent and changing conditions. Taking the system level reconfigurations into account, all manufacturing systems are configurable, although only reconfigurable machines have the ability of machine level reconfigurations. Therefore, a reconfiguration study should be carried out at both machine level and system level for the vast majority of manufacturing systems.

For most manufacturers, competition is stiff and margins are thin. Assuring that manufacturing assets are producing at maximum efficiency can mean the difference between success and failure. Manufacturers face two fundamental questions when making reconfiguration decisions: when to reconfigure and how to reconfigure. The objective of reconfiguration in manufacturing industry is to maximise the profit during each step of reconfigurations. Manufacturing optimisation distils down the reconfiguration decisions and enables the manufacturing reconfiguration process to be simpler, faster and more cost efficient.

1.2 Research context

The main characteristic features of today's dynamic manufacturing environments can be listed as follows: stochastic demand, variable but smaller production batch size, frequent and unpredictable changes in product mix, highly changeable processing, variable production sequences, very high volume of information and strong market competition. It has been most widely accepted that the industrial era dominated by mass production manufacturing is beginning to lose momentum and leading to a change. The global economy will be characterized by continuous innovation that will reward rapid product creation and development, and increased speed-to-market.

In current business environment, manufacturing companies are determined to develop and renovate by new standards and new customer satisfaction. Quality products are customized, designed and configured at the time of order. Products could be reconfigured and upgraded to meet evolving requirements, extending product life and reducing the value of distinct products. To become competitive and thrive under such environments, to be able to respond quickly and cost effectively to the global market, many appropriate businesses strategies are available for modern manufacturers:

1) Adapt new manufacturing technologies such as computer integrated manufacturing, reconfigurable manufacturing, etc.
2) Continuously improve the quality of the manufacturing by constantly advancing the criteria by which quality is measured. Extend and amplify customer satisfaction throughout the company. Identify infrastructure requirements that will enhance distributed concurrent product design, development and manufacture.
3) Create an environment of cooperation both within the company and with external technology and science bases. Work with existing consortia and other professional groups to promote and develop cooperative mechanisms.

4) Develop supplier-vendor-customer networks, incorporating interactive information exchange systems as appropriate.
5) Involve the work force in setting company agendas and in exercising initiative to accomplish them. Develop schemes that will measure the value of all level staffs as a corporate asset. Use these schemes to encourage continuous training and education.
6) Enforce laws and regulations on the environmental and social impacts of manufacturing, energy usage and conservation, workplace safety, and work force constitution.
7) Develop environmentally conscious manufacturing processes in collaborator with research centre.

The transformation of manufacturing that is underway is a dynamic process, one that will be shaped in part by unforeseeable developments. Reconfigurable manufacturing is a new paradigm, however a leading manufacturing technique which is designed to satisfy market demand and management goals by making better use of resources during the reconfiguration.

There are many aspects of reconfiguration, such as system reconfiguration, software reconfiguration, control reconfiguration, machine reconfiguration and process reconfiguration (Mehrabi, 2000). These also include various configurations of the production system e.g., serial, parallel and hybrid, reconfiguration of the factory communication software, configuration of new machine controllers, building blocks and configuration of modular machines, modular processes, and modular tooling.

All these reconfiguration methods can be developed and implemented to achieve the goals of manufacturer enterprises. For example, instead of 'make to order' which is by far the most preferable from the standpoint of 'return on capital employed', inventory holdings and cash flow, 'make to stock' can be considered to increase the utilisation of system capacity. Make to stock is often a replenishment exercise based on the days or weeks sales. To offer the maximum flexibility to a manufacturer, the ability to build the product based on previous order or at the time of the customer order is ideal in many environments. However, companies have to carefully assess the risk of carrying stock. Extending working hours is another popular choice for extra manufacturing capacity when it is feasible. It is often used in highly automated manufacturing environment where extra workforce cost for extending working hours is low.

In the process of designing and operating reconfiguration of manufacturing systems, one has to distinguish system level reconfiguration with machine level methods. At the system level, there could be several system configurations for production of the same part family. Development of the necessary tools and methodologies to design the system and evaluate various configurations (based on life-cycle economics, quality, system reliability, preferences of decision makers) is needed.

Development of a unified approach for capacity increase is an important challenge in the process of manufacturing reconfiguration. The increase of system capacity has direct effect on the system output and business profit which in term makes it a popular choice of reconfiguration decisions. Like any other reconfiguration problem, capacity change should be made among certain variables such as rump up time, layout requirement and cost. The decision making for this process is, however, quite complex since the number of variables is large. This thesis concentrates on the development of methodology for reconfiguring manufacturing systems by increasing capacity.

1.3 Aim and objectives

The aim of this investigation is to design and construct a simulation model for the reconfiguration of a manufacturing system. The outcome of the research is a methodology to optimize the system reconfiguration.

The primary research objectives of this investigation are given as follows:

- To design and construct a simulation model for a medium size manufacturing system
- To investigate machine reconfiguration options
- To optimise the system reconfigurations using several optimisation algorithms and to assess their performance
- To design and build reconfiguration cost model for cost implication analysis
- To study the use of product portfolio and its impact on manufacturing strategy

A robust simulation model is vital for this research as all the reconfiguration experiments are based on the model and the data it generates. Prior to building this model, a simple classic four machines model was built to study the WITNESS simulation language and obtain a better understanding of manufacturing simulation model. Thereafter, a medium size twelve machine (12M) manufacturing model was constructed and validated on the WITNESS simulation platform.

Reconfiguration can be carried out on both machine level and system level. The machine level configuration requires reconfigurable machine which is a US Reconfigurable Manufacturing System Centre in its own right (Engineering Research Center for Reconfigurable Machining Systems, 2001). The research provides options for both machine level and system level configuration depending on the gap between system throughput and market demand. The machine level reconfigurations can be simulated as the machine cycle time changes as it is the direct result of machine level reconfigurations.

The manufacturing system has a certain amount of complexity, twenty parts (20P) are processed on multiple machines before assembly. The reconfiguration decision is in fact a combinatorial optimisation problem. The research uses built-in optimisation algorithms in WITNESS to tackle the combinatorial problem and obtain the optimised reconfiguration solution.

Reconfigurations have to be carried out on a cost efficient manner as the basic aim of manufacturing companies is to make a profit. Using a reconfiguration cost model to calculate the total reconfiguration cost before making the reconfiguration decision will reduce the investment risk significantly. Cost models are built on stochastic data, in practice, real data should be used in order to achieve the right result.

Reconfiguration is not the only solution to meet market demand. If an investment assessment fails to meet the criteria, other solutions such as product portfolio restructuring will have to be considered by decision makers.

1.4 Research methodology

To study how manufacturing systems should be reconfigured and when to process the reconfiguration, a simulation approach is employed. There are many issues to be addressed in the research to answer the two questions about manufacturing reconfigurations. The goal is to use simulation results for reconfiguration analysis and decision making in order to maximize profit.

A medium size manufacturing model is used to simulate the manufacturing environment for reconfiguration. The simulation covers a group of machine producing a family of products for a specified number of production periods. It is important to realize that this approach is adopted for a simplified manufacturing environment under certain assumptions. One cannot, for example, assume that the best reconfiguration decision or strategy for this particular manufacturing system is also the best for another system.

The simulation experimental design is implemented via three steps to obtain the information for reconfiguration decisions. The simulation methodology developed here is the main contribution of the research which provides guidelines on how to use a simulation tool to aid decision making on the reconfiguration of manufacturing system. These steps are illustrated in Figure 1.1 and described as follows:

1. *Development of a manufacturing system model using WITNESS simulation language.*
   Manufacturing system data is randomly generated within a reasonable range to present a medium size manufacturing system. Exchanging this part of data will allow a company to simulate its real life system. Basic system performance analysis such as bottleneck identification, production scheduling and establishment of the original configuration is also carried out.
2. *Generation of market demand data based on Product life cycle.*
   Searching among the reconfiguration options, the results which reconfigure the system to satisfy the market demand is stored for optimisation. If the new reconfiguration still cannot meet the market demand, product portfolio restructuring should be considered in order to redesign the manufacturing system.
3. *Optimisation of the system output and analysis of the reconfiguration options.*
   Due to the great complexity of the system, optimisation algorithms such as simulated annealing and hill climbing are applied to optimise the manufacturing reconfigurations.

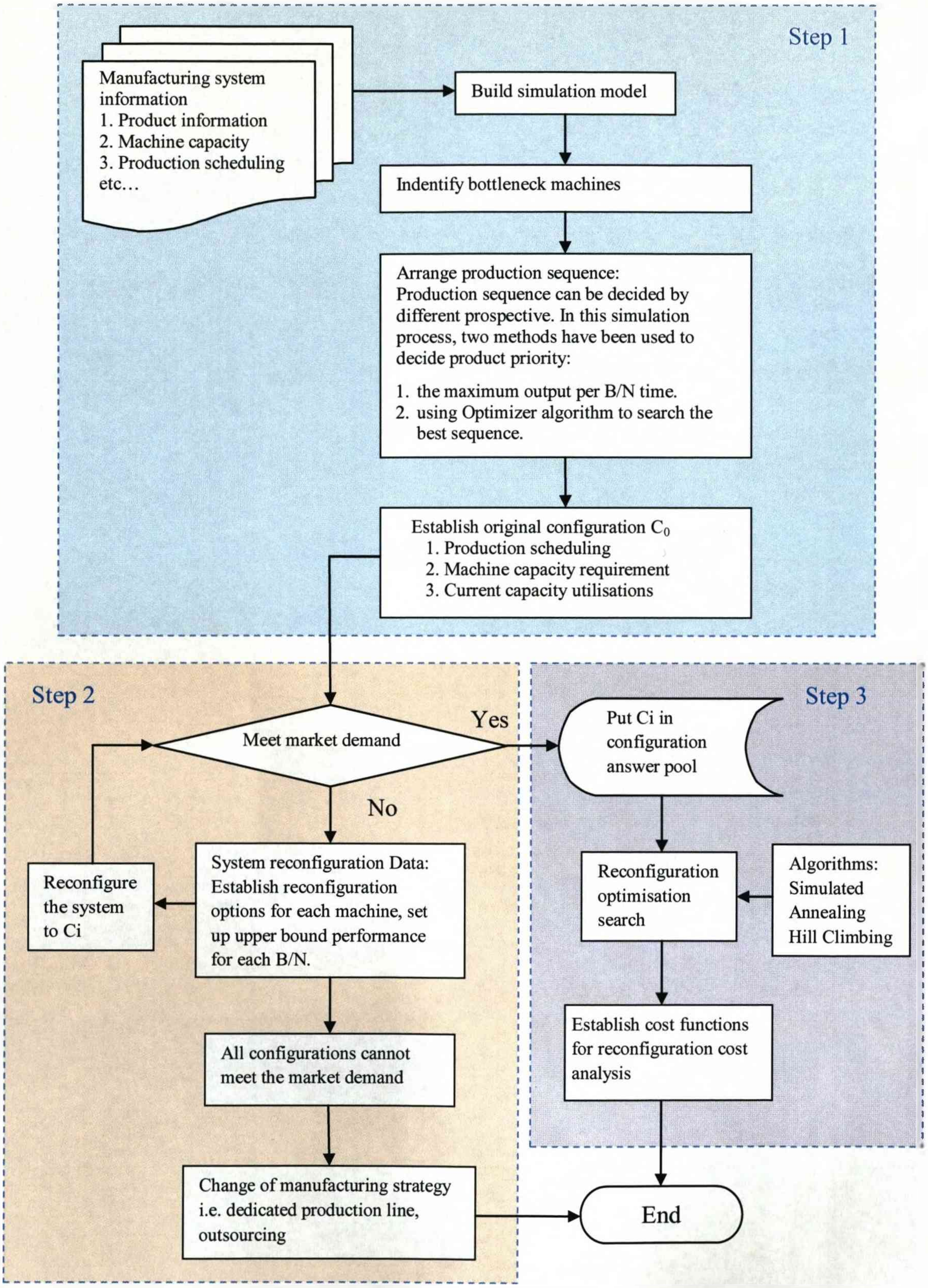


Figure 1.1: Flow chart for research methodology.

1.5 Structure of thesis

The remainder of this thesis is divided into seven chapters. These chapters are structured as follows:

Chapter 2 presents a literature review on traditional and reconfigurable manufacturing systems and previous work on the reconfiguration of manufacturing systems. Product life cycle, theory of constraints and simulation optimisations are reviewed in this chapter as they will be used in this research. In addition, a literature review on the cost of reconfiguration is presented and discussed.

Chapter 3 presents the simulation model which is an essential part of this research. It introduces the structure of the manufacturing model and simulation language used for building the model. For every aspects of the model, descriptions are given, assumptions are stated and variables are defined.

Chapter 4 describes the development of reconfiguration options which can be simulated in the manufacturing model to obtain the reconfiguration data. Questions on how to use Product Life Cycle to generate demand pattern, how to apply Theory of Constraints to identify the bottlenecks and how to calculate product priority to define the scheduling rules are all explained and elucidated here.

Chapter 5 presents the simulation optimisation experiments employed for the research. From the six built in optimisation algorithms simulated annealing is the main algorithm used and its ability to obtain the best results was also proved in this investigation. A comparison between simulated annealing and hill climbing algorithm and impact of parameter step size are discussed in this chapter.

Chapter 6 proposes and discusses a cost function for the manufacturing reconfiguration cost. The cost function separates the hard configuration cost with the soft cost which makes it possible to conduct the machine configuration cost directly from the simulation results. A group of linear and non-linear cost models are available as optional choice for decision makers.

Chapter 7 discusses more options on product portfolio when the market demand cannot be met easily by cost efficient machine reconfigurations. The launch of a new dedicated production line for high volume product and outsourcing are two strategies tested here.

Chapter 8 concludes the research on the reconfiguration of manufacturing system. In addition, the contributions of the research are summarised and future research directions are proposed.

Chapter 2
Literature Review

2.1 Introduction

Global economic competition and rapid social and technological changes have forced manufacturing to face a new economic objective: manufacturing responsiveness (Setchi & Lagos, 2004). Reconfigurability is defined as the ability to change and rearrange the components of a system repeatedly in a cost effective way. This chapter presents a state-of-the-art review on the reconfiguration of manufacturing system and simulation technology to conduct the reconfigurations. The investigation encompasses a number of technical areas that include the analysis of reconfigurable manufacturing systems. In this chapter, the reconfigurable manufacturing system and the theory which contributes to reconfiguration decision making are reviewed and are divided into the following sections:

1. The reconfiguration of manufacturing systems
2. Theory of Constraints
3. Simulation of manufacturing system
4. Scheduling and planning of manufacturing system
5. Optimisation of manufacturing system reconfiguration

2.2 The reconfiguration of manufacturing system

2.2.1 Rationale of reconfigurable manufacturing system

Globalisation has created a new landscape for manufacturing, leading to shortened product life cycles, shortened windows of market opportunity and frequent changes in product demand. This change presents both a threat and an opportunity. Firms must learn to operate effectively in a dynamic production environment characterized by increasingly unpredictable market demands and the proliferation of product variety, as well as rapid changes of product and process technologies (Westkämper E, 2000).

To capitalise on the opportunity, manufacturers need to have the systems that can produce a wide range of products within a product family at the required quantity. The product range and quantity must meet the requirements of multiple countries and various cultures, not just one regional market. The Reconfigurable Manufacturing System (RMS) has the capabilities that allow for quick changeover of product mix and quantities that might vary dramatically, even on a monthly basis. An RMS is assumed not to be more expensive than flexible manufacturing systems or even dedicated transfer lines. Unlike other types of manufacturing systems, an RMS is installed with the exact production capacities and functionality needed, and may be upgraded in the future, whenever needed (Mehrabi, 2000). Reconfigurable manufacturing systems (RMS) have been widely recognized as one of the promising key technologies offering a competitive edge in the new manufacturing era (Koren, 1999).

2.2.2 Definition of reconfigurable manufacturing system

A reconfigurable manufacturing system (RMS) is one designed at the outset for rapid change in its structure, as well as its hardware and software components, in order to adjust its production capacity and functionality quickly within a part family in response to sudden market changes or in regulatory requirements (Koren, 1999). Reconfigurable manufacturing systems (RMS) are a new class of manufacturing systems aiming at combining the high throughput of dedicated manufacturing lines with the flexibility of flexible manufacturing systems (Jun Du, 2006).

The RMS as well as one of its components – the Reconfigurable Machine Tool (RMT) were invented and patented in 1999 in the Engineering Research Centre for Reconfigurable Manufacturing System at the University of Michigan. The RMS goal is summarised by the statement 'Exactly the capacity and functionality needed, exactly when needed' (Koren & Kota, 1999).

2.2.3 Key characteristics of reconfigurable manufacturing system

Five core RMS characteristics have been proposed by Koren and followed by researchers in the reconfigurable manufacturing area to design and implement an ideal reconfigurable manufacturing system. These core characteristics are (Koren, 1999):

Modularity: All major components of RMS are modular (e.g., structural elements, axes, control, software and tooling).

Integrability: Machine and control modules are designed with interfaces for component integration.

Customisation: To reduce system cost, the machine and controller configuration must be customised to fit the dominant features of a part family and the application by utilising the concepts of customised flexibility and customised control.

Convertibility: Short conversion time between different production batches is a major requirement. In order to achieve this, rapid tuning of the tools, raw material, software and fixtures has to take place.

Diagnosability: Detecting unacceptable part quality is critical in reducing ramp-up time in RMS. As production systems become more reconfigurable and are modified more frequently, it is essential to tune rapidly a newly reconfigured system so that it produces quality parts.

Another important factor labelled as Scalability was recently discussed (Koren, 2006). There are similarities between scalability and convertibility but whereas the latter is concerned with the machine level, scalability operates more at the system level.

Scalability: The ability to change production capacity easily by rearranging an existing manufacturing system and/or changing the production capacity of reconfigurable stations. Scalability is the counterpart characteristic of convertibility. Scalability may require adding spindles to a machine to increase its productivity at the machine level, and at the system level changing part routing or adding machines to expand the overall system capacity as the market for the product grows.

A typical RMS will have several of these characteristics, though not necessarily all. With such characteristics, an RMS increases the speed of responsiveness to unpredicted events, such as sudden market demand changes or unexpected machine failures.

2.2.4 Comparison of manufacturing systems – DML, FMS and RMS

The new paradigm required by today's manufacturing should not only incorporate the advantages of Flexible Manufacturing Systems (FMS) but also be simpler, responsive, and less costly. The Reconfigurable Manufacturing Systems (RMS) paradigm attempts to satisfy these requirements and avoid the shortcomings of previous conventional manufacturing philosophies (Setchi & Lagos, 2004). DML (Dedicated manufacturing lines), FMS and RMS can have similar system configurations. Table 2.1 summarises the three major types of manufacturing systems and their definitions.

Table 2.1 Summary of three types of manufacturing systems (Mehrabi, 2000).

Manufacturing systems	Definitions & objectives
Dedicated manufacturing lines (DMLs)	A machining system designed for production of a specific part at high volume. Cost-effectiveness is the objective achieved through pre-planning and optimisation.
Flexible manufacturing systems (FMSs)	A Flexible Manufacturing System is an integrated system of machine modules and material handling equipment under computer control for the automatic random processing of palletised parts. The objective is to manufacture cost-effectively a family of parts that can change over time, with minimum changeover cost, on the same system at the required volume and quality.
Reconfigurable manufacturing systems (RMSs)	A Reconfigurable Manufacturing System is designed for rapid change in structure in order to adjust production capacity and functionality quickly, within a part family, in response to changes in marker requirements. The objective is to provide exactly the functionality and capacity that is needed, when it is needed.

Traditional manufacturing systems such as DMLs or cellular manufacturing systems (CMSs) cannot cope with the characteristics of the ever-changing global market. Even FMSs cannot deal with these modern challenges in a cost-effective manner. RMSs aim at combining the high throughput of a DML with the flexibility of an

FMS, maintaining the ability to deal with a variety of products and volumes in a cost-effective manner (ElMaraghy, 2007).

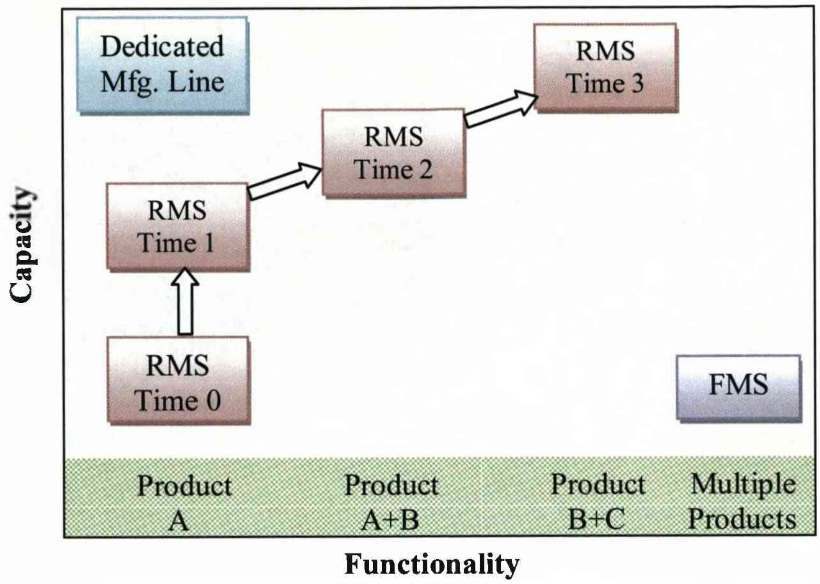


Figure 2.1: Both DML and FMS are static systems, while an RMS is a dynamic, evolving system (Koren, 1999).

The history of manufacturing systems shows their evolution over the years in response to an increasingly dynamic and global market with greater need for flexibility and responsiveness (Youssef and ElMaraghy, 2006). As shown in figure 2.1, RMS may lie between a DMS and a FMS in terms of capacity and functionality. The key characteristic of RMS is that, unlike a DMS and a FMS, its capacity and functionality are not fixed. The FMS priority is built-in flexibility above all other features. It is robust but has high initial capital investment cost and often underused (Mehrabi, 2000).

The RMS concept has emerged in the last few years in an attempt to achieve changeable functionality and scalable capacity (Koren, 1999). A complete reconfigurable manufacturing system does not yet exist but is the subject of major research efforts around the world.

2.2.5 Manufacturing system reconfiguration

Since a manufacturing system can be upgraded by using hardware and software modules that can be integrated quickly and reliably, it is critical to plan the appropriate reconfiguration on the system level and its components. Reconfiguration requires strategic decisions that determine if machines need to be added or removed, if the system layout and flow paths need to be changed at the lowest cost (Li and Koren, 2006).

It is known that any manufacturing system is able to have reconfigurability and responsiveness ability without considering system cost (Koren and Weber, 1998). However, system cost is most important issue among all manufacturing system questions and it must be considered. The reconfiguration process can be carried out on any manufacturing system. The major advantage of RMS is to scale the capacity not only at the system level but also at the machine level by virtue of its modular and structures.

Most of the previous studies are limited to the reconfiguration design for a new RMS system. The problem of manufacturing system reconfiguration for an existing one during its active period was often overlooked in the literature. Tang and Koren present the models for the reconfigurability of a RMS as a network of potential activates and configurations to which a shortest path strategy is applied (Li and Koren, 2006). Youssff proposed a model for optimizing the cost of RMS configurations with multiple aspects using Genetic Algorithm (Youssff, 2006). Bradford and Childe studied a nonlinear redesign methodology for manufacturing systems which uses an iterative strategic model to optimizing configuration for continuous change (Bradford & Childe, 2002).

Saad and Gindy approached manufacturing reconfiguration in a different way. The integration of production planning, process planning, loading, scheduling and cell creation is considered to achieve reconfiguration. The production requirements of products and the capabilities of manufacturing cells are defined by generic capability units, which are known as resource elements (Saad & Gindy, 1999).

A multiple objective Tabu search based adaptive simulation optimisation system is used for loading and scheduling the existing cells for the current production requirements which are obtained from the production planning system at the beginning of each loading period. If the performance indicators are unsatisfactory for the coming period then the reconfiguration (creation of new cells) is considered. This could be an answer for the first question ''when do we need to reconfigure?'' In the proposed integrated approach, the reconfiguration action is performed by a multiple objective tabu search based adaptive simulation optimisation model which generates possible virtual cell configurations and selects the one which satisfies the desired performance levels. This can be considered as an answer for the second question ‘‘how do we reconfigure?’’ (Saad, 2003). In their recent studies, the reconfiguration of manufacturing systems is discussed and an integrated framework which is mainly based on multiple objective simulation optimisation is proposed for reconfiguration of manufacturing cells (Saad, Baykasoglu and Gindy, 2002).

Toguyeni implemented knowledge-based techniques, considering two different levels of reconfiguration. The former, called minor reconfiguration, considers only an operational reconfiguration for preventing taking a control sequence that requires out-of-order operation performed by a resource in faults. The latter, referred to as major reconfiguration, takes into account all of the potentialities of the production architecture (Toguyeni et.al., 2003).

Deif and ElMaraghy presented an approach to model the capacity scalability planning. The effect of the reconfiguration costs on the capacity scalability planning horizon and overall cost is investigated. The results showed the relation between deciding on the optimal capacity scalability planning horizon and the different reconfiguration costs (Deif & ElMaraghy, 2006).

Hon and Xu addressed the complex relationship between the product life cycle for a family of products and the manufacturing systems performance optimisation via reconfiguration. The paper using a simulation model to explain the proposed the reconfiguration process and the results show the impact of product life cycle on manufacturing reconfiguration (Hon & Xu, 2007).

The focus of this research addresses when, where, and how much should the capacity of the manufacturing system be scaled and it can be applied to nearly any manufacturing systems.

2.2.6 Cost of reconfiguration

A configuration path is used to describe the set of system configurations that a RMS assumes as it changes over time (Patrick & Carlo, 2007). Numerous manufacturing cost models have been developed based on traditional accounting methods. Son proposed a similarity-based reconfiguration cost model in his PhD dissertation (Son, 2000). Youseef and ElMaraghy presented a related metric for assessing reconfiguration smoothness (Youssef, 2005). Patrick Spicer illustrated a thorough mathematical model to compute the reconfiguration cost between two system configurations (Patrick & Carlo, 2007). The cost model added physical arrangement cost with lost capacity cost at ramp-up time to calculate the total reconfiguration cost.

Deif and ElMaraghy solved the capacity expansion problem for RMS assuming that the reconfiguration cost was constant (Deif and ElMaraghy, 2005). More recently Deif and ElMaraghy presented a cost function to aid the system' designers in deciding how much to scale the systems in order to meet the market demand in a cost-effective way (Deif & Elmaraghy, 2007). Their cost function is shown in the following equation:

$$C = \sum_{t=1}^{n} C_t(v_t) + \sum_{i=1}^{n} CR_i \qquad [2.1]$$

Where,

C = *the cost of reconfiguration.*

$C_t(v_t)$ = *the cost of the physical capacity unit that the system will be scaled with at the time period t.*

CR_i = *other costs of reconfiguration that are associated with the scaling.*

n = *the number of capacity scalability points and* $n<T$.

t = *Time is idealized to be consisted of discrete periods* $1,2,\ldots,T$.

The function represents the cost of having a capacity level v. It is time-dependent and is expressed in terms of the present value of costs as of time1. This cost function is composed of two components, the first reflects the cost of the physical capacity unit $C(v)$, and the second represents the cost associated with this physical scaling or reconfiguration of the system CR. Thus, the cost for each period t is mainly the cost of having a capacity to satisfy the demand. On the other hand, the term CR_i represents other costs of reconfiguration that are associated with this scaling, and basically includes other related cost parameters, such as the cost of downtime to rescale the system or to ramp up the new configuration with the new capacity, the labour cost involved and the effort required for that reconfiguration or scaling.

Each cost model defines and categorises the various costs during the reconfiguration process and sums them to get an overall reconfiguration cost. The cost will have to be estimated before the reconfiguration to aid the decision of when to reconfigure and the level of reconfiguration. In order to make a good decision, every part of the reconfiguration cost must be taken into consideration.

2.3 Theory of Constraints

The Theory of Constraints is the invention of Eliyahu Goldratt. It is a business philosophy which seeks to strive towards a global objective, or goal of a system through an understanding of the underlying cause and effect dependency and variation of the system in question. Since *The Goal* first appeared in 1984, the ideas introduced have drawn a wide range of responses from many parts of the manufacturing and academic world (Goldratt, 1984).

Theory of Constraints (TOC) is an example of a management philosophy built upon a limited number of assumptions and designed to provide a process of continuous ongoing improvement (Sivasubramanian, 2003). According to TOC, every organisation's outputs are determined by its constraint(s). If the system is a for-profit business, of which including manufacturing, then the goal is to make more money, both now and in future. Throughput, inventory, and operating expense are measures used to assess performance toward this goal.

The TOC is a theory which deals with bottlenecks in a manufacturing system. A bottleneck is a phenomenon where the performance or capacity of an entire system is limited by this single component. This component is sometimes called a bottleneck point. A bottleneck lies on a system's critical path and provides the lowest throughput. Bottlenecks are usually avoided by system designers. In addition a great amount of effort is directed at locating and tuning them.

2.3.1 Five focusing steps

The Theory of Constraints is based on the principle that the rate of revenue generation is limited by at least one constraining process – a bottleneck. Only by increasing throughput at the bottleneck process can overall throughput be increased.

The key steps in implementing an effective TOC approach are (Goldratt, 1990):

1. Identify the constraint (bottleneck)
2. Decide how to exploit the constraint

3. Subordinate all other processes to the above decision
4. Elevate the constraint
5. If, as a result of these steps, the constraint has moved, return to step 1.

These five key steps are an ongoing process of improving a system. In this research, finding the bottleneck machines and the steps required to reconfigure the bottleneck machines is based on the philosophy of TOC.

2.4 Manufacturing Simulation

Simulation is the imitation of the operation of a real-world process or system over time. Simulation involves the generation of an artificial history of the system and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system that is represented (Banks, 1999).

Using simulation to describe and analyse the behaviour of manufacturing system, asking 'what if' questions about the real system, simulation is a powerful problem-solving methodology for aiding decision making in manufacturing industry. For instance, manufacturers can use simulation to draw conclusions about the benefits and risks associated with introducing a new product, changing current scheduling rules, adding new machines, using new production technology and outsourcing parts.

Murphy and Perera stated that simulation modelling is used to design and experiment numerous scenarios that are refined before being put into physical practice within business environment (Murphy & Perera, 2001). In order to gain more benefits from simulation, companies need to plan the whole modelling procedure in detail. As no simulation provides 100% realistic duplication of the actual process, what information can be collected and will be embedded in the model is very critical for simulation projects. Computer simulations can generate large amounts of data that can lead to a false sense of security in the numbers (Ball, 2001). In this research, the primary goal is to provide knowledge, or answers to important questions through experimentation and modelling. The information in the simulation model has been kept simple without making it too difficult to obtain.

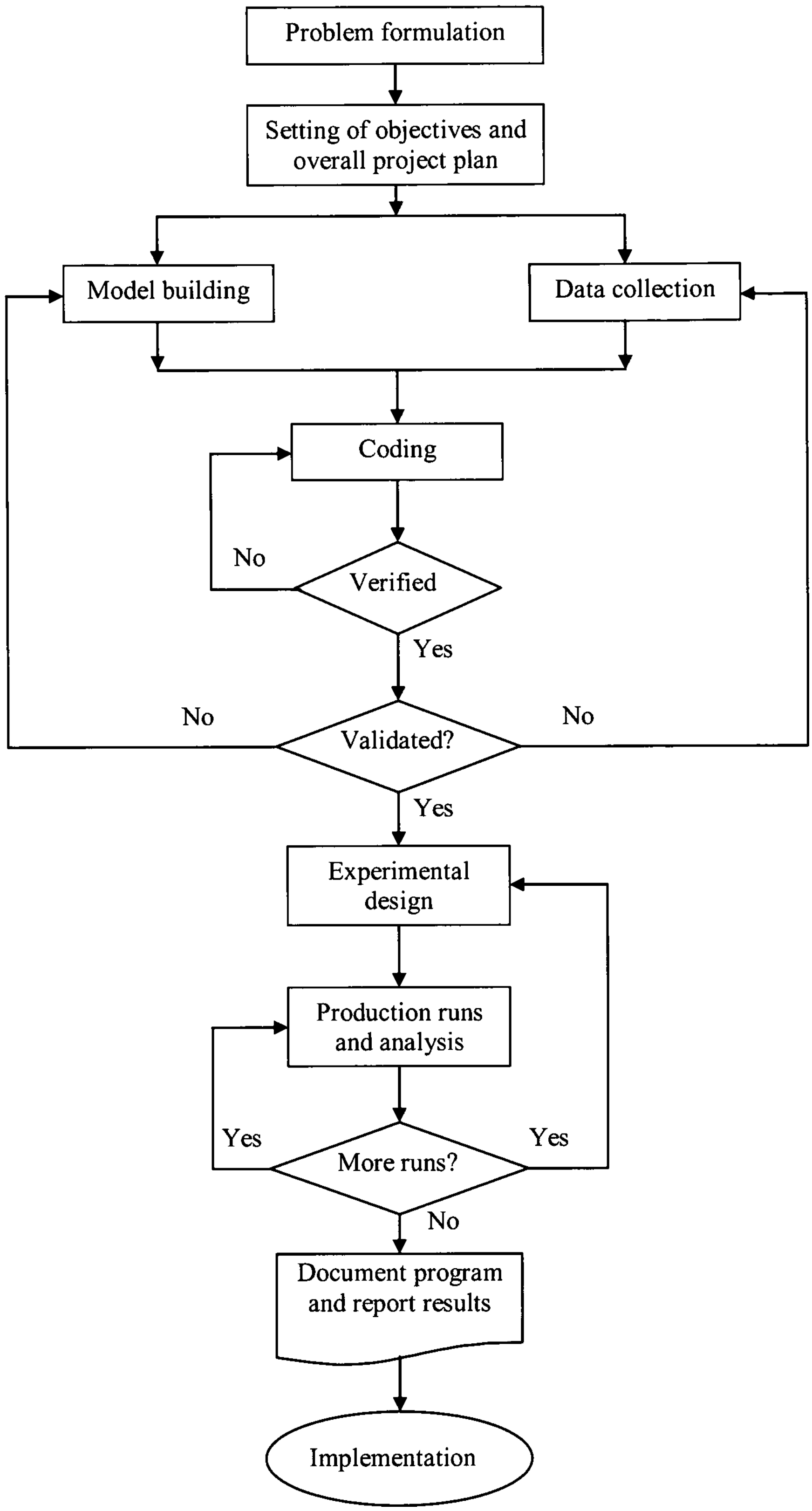


Figure 2.2: Steps in a Simulation Study (Banks, Carson, Nelson, & Nicol, 2000).

2.4.1 Discrete Event Simulation Model

Simulation can be used for a vast spectrum, from inner mechanisms of atoms to the behaviour of the universe. Manufacturing systems with their intrinsic complexity are certainly good areas for simulations. Simulation, the discrete-event simulation (DES) in particular, has been applied to various aspects of manufacturing since the 1960s (Law and Kelton, 2000).

A discrete-event simulation model is defined as one in which the state variables change only at those discrete points in time at which events occur. Each event occurs at an instant in time and marks a change of state in the system. Discrete event simulation models are bonded with other types of models such as mathematical models, descriptive models, statistical models and input-output models (Banks, 1999).

Manufacturing engineers use Discrete Event Simulation (DES) to respond to the changing of global market. Discrete Event Simulation of manufacturing systems has become widely accepted as an important tool to aid the design and decision making in manufacturing domain. Numerous papers focus on shortening the time to market (Terwiesch, 2001; Mansurov, 2001; Driva, 2000) and dealing with shorter product life cycles (Driva, 2000). Smith described the use of a discrete event simulation for controlling a flexible manufacturing system (Smith, 1994). They seek to use the same simulation model for system design, analysis and control. Alexander considered discrete event simulation as a method of evaluating the overall batch cycle time including interactions over time (Alexander, 2006). Johansson and Kaiser examined to what extent DES can be applied to the evaluation of resetting performance in manufacturing systems (Johansson & Kaiser, 2002). The case study of a DES model of a factory unit in Sweden proved that the DES could be used for the evaluation of resetting process in manufacturing systems. Faget used discrete event models developed by Toyota Motor Company as a method for detecting bottlenecks (Faget, Eriksson, & Herrmann, 2005). The bottleneck detection method has also been integrated into MS Excel spreadsheets for easy use by the decision makers without knowledge of simulation.

Although DES is recognized as a successful tool in the industry, there are still some problems of implementation and the levels of reliability. Johansson pointed out that due to inadequate practices within the organisation; users often build simulation models with estimated data (Johansson, et al. 2003). It is very difficult to gather enough reliable manufacturing data. Moreover, DES is used for solving one-of-a-kind problems rather than for the same task in every new development project (Williams, 1996). The problem Klingstam found in DES projects in Sweden and Johansson's survey of DES both pointed out the limitation of discrete event simulation which is a stand-alone-technique rather than an integrated part of the development process (Klingstam 2001, Johansson, et al. 2003).

2.4.2 Simulation applications in manufacturing systems

For many decades manufacturers employ simulation to develop descriptive computer models and exercise those models to aid manufacturing system design and operation. Two primary computer simulation books give a comprehensive and extensive introduction of simulation by Banks, Law and Kelton (Banks 1998, Law and Kelton, 2000). Additionally, three examples of conferences are the Winter Simulation Conference (WSC) held in December at www.wintersim.org, the Summer Computer Simulation Conference typically held in July and sponsored by the Society for Modelling and Simulation International at www.scs.org, and the Simulation Solutions Conference usually held in the spring at www.simsol.org. They are organized exclusively for simulation technology, languages and applications. Numerous publications contribute the development of this significant tool in general systems analysis and particular manufacturing systems design and operation.

2.4.2.1 System design and layout

Manufacturing system design includes facility design, materials handing design, manufacturing cell design, FMS and RMS system design. There are a vast amount of research studies on manufacturing system design and redesign using simulation.

Savsar described the use of simulation to generate and evaluate alternative layouts as part of a general layout design procedure (Savory, 1991). Aly and Subramaniam developed a simulation based decision support system for the FMS design problem. The decision support system included a multi-attribute utility model to reflect the company policy and also various performance measures at different levels. Different designs stage needs different decision-making strategy (Aly, 1993).

Williams and Gevaert described the analysis of a production system at an automotive supply company (Williams, 1997). Williams and Celik analysed an automobile final assembly line using simulation. The simulation model was based on proposed configuration and operational information. The model was used to verify if the proposed system would meet required production rates and to predict the relative performance of alternative configurations (Williams and Celik, 1998).

Taj et al. used simulation to verify the design of manufacturing cells. The simulation models were used to ensure that the selected designs could meet market demand (Taj, Cochran, & Duda, 1998). Nandkeolyar, Ahmed and Pai illustrated the use of a simulation model to predict the performance of a manufacturing cell for the manufacturing of hydraulic flow control components. The model was used to provide justification for a reconfiguration of a traditional facility into a cell-based system (Nandkeolyar, Ahmed, & and Pai, 1998).

Bozer and Kim used simulation to evaluate the performance of analytical models aimed at determining the optimal or near optimal transfer batch sizes in manufacturing systems. As part of this work, they also developed analytical relationships between materials handling capacity and expected work in progress (Bozer and Kim, 1996).

2.4.2.2 Manufacturing system operations

Applications of simulation on manufacturing system operation generally are considered with making shorter-term decisions when compared to the system design application. In this area, operations planning and scheduling (Cf. Section 2.3), real-time control, operating policies, and performance analysis are the major research interest.

Harmonosky provided a review of real-time scheduling research. The majority of the reported developments involved the use of simulation as either an evaluation or control mechanism (Harmonosky, 1995). Harmonosky and Robohn provided information on computer time required to make simulation runs of physical manufacturing systems. The results of this analysis provided a foundation for evaluating the appropriateness of using simulation for real-time control for different types of systems (Harmonosky, 1995). Byrne and Chutima used a simulation model of an eight-machine FMS to evaluate the performance of a proposed real-time control policy (Byrne, 1997).

Iyer and Askin illustrated the development and use of a general-purpose simulator to evaluate operating policies for manufacturing cells which simultaneously considered the configuration of system resources, part mix, worker assignment rules, and part dispatching rules. Simulation was identified as the only viable analysis methodology given the strong interactions between these entities (Iyer, 1998).

Bai used simulation to analyse the performance of an example production system under different control policies. The experiment considered five input control strategies and four dispatching rules for a three-machine, single product system. The objective of the research was to evaluate the feasibility of using simulation for production control policy selection (Bai, et al. 1996).

Park, Matson, and Miller presented a simulation model used to verify that daily throughput requirements could be met in a new Mercedes-Benz assembly plant. Moreover, the simulation model was used to determine the maximum throughput of the facility and characterised how the component buffers behaved in terms of quantity fluctuations and identified possible system bottlenecks (Park, 1998).

2.4.3 Simulation languages and packages

Generally speaking, computer simulation is and has always been an expensive endeavour. However, the use of simulation in manufacturing has been driven to a large extent by the increases in computational power over the past decades. The main costs involved in the adaptation of simulation are computers, software, programmer and data collection cost. With the development of user friendly software, the popularity and importance of simulation increases constantly.

Simulation software surveys provide a source of information that periodically appear in professional society journals. For example, OR/MS Today (www.lionhrtpub.com), a publication of Institute for Operations Research and the Management Sciences, reported a simulation software survey in December 2005, which has since been updated on the website. Baldwin et al. presented a survey on the use of simulation software and further improvement (Baldwin, et al. 2000). They concluded that current simulation packages were easy to use, visual effective and interactive but limited for complex and non-standard problems and were slow. Swain compared 39 organisations for the development and application of simulation software packages (Swain, 2001). Table 2.2 shows a list of available Commercial-off-the-shelf COTS simulation modelling packages and their vendors.

Table 2.2: Simulation software, vendors and typical applications.

Simulation Package	Vendor	Typical Applications of the software
AnyLogic	XJ Technologies	Forecasting and strategic planning, process analysis and optimization, optimal operational management, process visualisation
Arena	Rockwell Software	Facility design/configuration, scheduling, passenger and baggage-handling processes, patient management, dispatching strategy
AutoMod	Brooks Automation	Discrete event simulation to improve the design, configuration and optimization of material handling processes
EXTEND Suite	Imagine That, Inc	Professional 3D modelling of continuous, discrete event and discrete rate processes.
FlexSim	Flexsim Software Products, Inc	Manufacturing, material handling, warehousing, supply chain, process improvement, lean, healthcare, continuous, food
GoldSim	Goldsim Technology Group, LLC	water resources, mining, hazardous waste management, probabilistic risk analysis, reliability and throughput analysis
GPSS/H/Proof Animation/SLX	Wolverine Software	Queuing models
Micro Saint Sharp	Micro Analysis & Design	Micro Saint Sharp is a powerful general-purpose discrete event simulation tool that allows users to build models of processes
ProModel	ProModel	Business process improvement - all areas
ShowFlow	Webb Systems Limited	simulation of material flow and more generally of process flow
Simul8	Simul8 Corporation	Optimize throughput, maximize resource utilization, identify bottlenecks, reduced risk decisions, process management, learning and training. For comprehensive, easy to build simulations requiring power features.
WITNESS	Lanner Group	Strategy validation, operational planning and process improvement

In general, the discrete event simulation function of all simulation software packages works in similar way. Schriber and Brunner gave a concise description of how discrete event simulation software works (Schriber and Brunner, 2001). One of the main difficulties in simulation is the selection of the correct package for a particular application. Hlupic's survey showed only 27.7% of simulation users used one simulation package. Simul8 and WITNESS are the top two simulation package used in the UK academia. However, it should be noted that nearly 85% of simulation projects were carried out in manufacturing areas (Hlupic, 2000).

Barton et al. summarised the simulation industry need and predicted the future in areas of education, research and software (Barton, Fishwick, Henriksen, Sargent, & Twomey, 2003). He also proposed an expression to present the primary goal of software as follows:

$$\frac{Functionality \times Ease-of-use}{Cost \times Complexity} \qquad [2.2]$$

Maximum the above expression is the focus of software development. A product can have all the functionality one could ever need but if its features are too hard to use, cost too much, or are unnecessarily complex, the product will fail.

2.5 Operations Planning and Scheduling

The two key problems in production scheduling are, according to Wight, 'priorities' and 'capacity'. In other words, 'What should be done first' and 'Who should do it'. Wight defined *scheduling* as 'establishing the timing for performing a task' and observed that, in manufacturing firms, there were multiple types of scheduling, including the detailed scheduling of a shop order that showed when each operation should start and complete (Wight, 1984). Cox defined operation scheduling as 'the actual assignment of starting and/or completion dates to operations or groups of operations to show when these must be done if the manufacturing order is to be completed on time.' (Cox, 1992).

The scheduling problem is most often described as sequencing n jobs on m machines in a way that a certain performance criterion is met. In recent years, a few researchers have proposed new algorithms or methodologies of process planning and scheduling in order to achieve superior overall system performance (Parunak, 1991). Scheduling is an optimisation process, and thus refers to one or more objectives. Typical scheduling objectives are (Tan & Khoshnevis, 2000):

- minimizing the average flow time
- minimizing the makspan
- minimizing the average tardiness
- minimizing the average work-in-process inventory
- maximizing the probability of meeting the due date
- maximizing equipment utilisation
- maximizing the throughput

Saad, Baykasoglu and Gindy proposed integrated system dealing with the multiple objective optimisations which takes into account the performance of operational levels before finalizing the loading and production schedule. In the case of a proposed integrated system unable to find a solution that satisfies the management's goal, a reconfiguration of system resources is carried out. If the problem still exists with no suitable solution, new resources should be added to the system. In this case the system would have a new configuration. Configuration and reconfiguration problems are addresses in their research (Saad, Baykasoglu and Gindy, 2002)

2.5.1 The difficulty of scheduling

The difficulty of the scheduling problem lies in its computationally hard nature. The computer time required for the optimal solution grows exponentially with the size of the problem. It is usually simple to state a scheduling problem; however it is one of the hardest combinatorial optimisation problems that could be encountered. Formally speaking, scheduling problem is NP-hard. An NP problem – "non-deterministically polynomial" – is one that, in the worst case, requires time poly-nominal in the length of the input for solution by a non-deterministic algorithm (Garey & Johnson, 1979).

Small scheduling problems can be solved quickly with many different methods. For example, minimizing of the total elapsed time with n jobs and 2 machines can be solved by Johnson's rule and minimizing the number of tardy jobs with m machines and 1 job in a flow-shop can be solved by using graphical techniques and Gantt Charts. There are $(n!)^m$ number of total possible solutions for a problem with n jobs and m machines. Therefore, for problems in which the number of machines is larger than three, no general closed form solution exists (Ferrell, Sale, Sams, & Yellamraju, 2000).

A real scheduling problem, in general, is further complicated by the dynamic shop floor environment in which new orders arrive on a frequent but unpredictable basis. In such situations, scheduling process must be dynamic and actuate. The research in scheduling has been primarily focused in the construction of efficient scheduling rules and heuristics. The development of computer simulation also contributes to the research in scheduling.

2.5.2 Scheduling rules

In the sequencing/scheduling literature, terms such as scheduling rule, dispatching rule, priority rule, or heuristic are often used synonymously (Iskander, 1977). Gere has made an attempt to distinguish between priority rules, heuristics, and scheduling rules. He considers priority rules as simply a technique by which a number (or value) is assigned to each waiting job according to some method and the job with minimum 'value' is selected. Gere defines a heuristic to be simply some 'rule of thumb', whereas a scheduling rule can consist of a combination of one or more priority rules and/or one or more heuristics (Gere, 1966). Job shop scheduling rules can be used to control when selected jobs run in relation to other jobs. The following scheduling rules are most popular in the literature.

1. Johnson's rule
2. First in first out (FIFO)
3. Shortest process time (SPT)
4. Earliest due date (EDD)
5. Critical ratio

Fundamentally speaking, scheduling rules give the criteria to be followed during the scheduling process. As mentioned earlier, scheduling rules use one or more manufacturing parameters to achieve one or more objectives. None of the rules are able to achieve objectives in every way due to the conflicting nature manufacturing parameters. For example, the shortest process time aims at maximizing the total throughput. It is impossible to minimizing the average flow time. None of the rules dominates in overall performance, i.e., one rule may be superior for a set of benchmark while another may be better for another set of criteria. This conclusion is supported by several surveys (Iskander 1977, Ferrell, et al. 2000).

2.5.3 Using simulation for scheduling problems

Baid and Nagarur describe the use of simulation as part of an integrated decision support system for FMS. They point out that 'simulation can contribute to the decision-making process at all three levels of (manufacturing) managerial planning-strategic, tactical, and operational levels' (Baid, 1994). Many researchers have proposed their solutions for scheduling problems. The main solution available in the literature can be categorised into the following sections.

Most scheduling problems are combinatorial optimisation problems which are too difficult to be solved optimally, and hence heuristics are used to obtain good solutions in a reasonable time. Park evaluated sixteen heuristics that were based on Johnson's algorithm (Park, 1988).

Laarhoven used simulated annealing technique for the problem of finding the minimum makespan in a job shop based on a randomised version of iterative improvement. The probabilistic element of the algorithm made simulated annealing a significantly better approach than the classical iterative improvement approach on which it is based (Laarhoven, Aarts, & Lenstra, 1992). Aldowaisan and Allahverdi proposed two heuristics based on Simulated Annealing and Genetic Algorithm for the no-wait flowshop problem to minimize makespan. Extensive computational experiments showed that the simulated annealing heuristic outperformed the best two existing heuristics (Aldowaisan & Allahverdi, 2003).

Many promising and efficient heuristics have been reported in the literature, most of them are capable of finding good quality solutions in a very short time. However, none of them dominates in overall performance, i.e., one heuristic may be superior for a set of benchmark test cases, while another may be better for another set of test cases (Tan & Khoshnevis, 2000).

2.6 Optimisation Algorithms

2.6.1 Description of optimisation problems

Simulation optimisation provides a structured approach to determine optimal input parameter values, where optimality is measured by a function of output variables associated with a simulation model.

Consider a discrete-event simulation model with p deterministic input parameters $\psi \equiv (\psi_1, \psi_2, \ldots, \psi_p,)$ and q stochastic output variables $Y \equiv (Y_1, Y_2, \ldots, Y_q)$, where Y is a function of ψ ($Y = Y(\psi)$). Suppose the input parameters are defined over a feasible region ψ. Define a real function of Y, C(Y) that combines the q output variables into a single stochastic output variable. The goal is to determine values for ψ such that $F(\psi)$, the simulation response function, is optimised (Swisher, Hyden, Jacobson, & Schruben, 2000).

The p deterministic input parameters ψ can be either continuous or discrete. The challenge associated with finding the best output Y is that $F(\psi)$ cannot be observed directly. It may require multiple simulation run replications or long simulation runs. For this reason, several metaheuristic optimisation algorithms (Cf. section 5.2) have emerged as efficient tools for solving such problems.

2.6.2 Most used optimisation algorithms

A number of algorithms are often used in simulation, e.g. Genetic Algorithm (GA), Simulated Annealing (SA), Ant Colony (AC), Hill Climbing (HC), and Tabu search (TS). Except the Hill Climbing which belongs to local optimisation, the others are global optimisation algorithms. Genetic Algorithm is a general-purpose stochastic and parallel search method based on the mechanism of natural selection and natural genetics. GA is a search method that has potential of obtaining near-global minimum. Simulated annealing is a powerful, general-purpose stochastic optimisation technique, which can theoretically converge asymptotically to a global optimum solution. TS is based on the hill-climbing method that evaluates iteratively a best solution each time the neighbourhood is updated. The neighbourhood around the initial solution is created to examine whether the initial solution allows creating better solutions (Blum & Roli, 2003). Ant colony was inspired by the behaviour of real ants, while almost blind, are capable of finding the shortest path from food sources to the nest. The process is characterized by a positive feedback loop, where the probability increases with the number of previous steps that chose the same path (Dorigo & Colorni, 1996).

This section will illustrate the processes of 'Simulated Annealing' and 'Hill Climbing' as they are the algorithms available in the simulation program used in this research.

2.6.2.1 Hill climbing

Hill climbing is an optimisation technique which belongs to the family of local search. It is a popular first choice as it is a relatively simple technique to implement. Although other above mentioned advanced algorithms may give better results, there are situations where hill climbing works well. Hill climbing can also operate on a continuous space. In that case, the algorithm is called gradient ascent or gradient descent if the function is minimized.

Hill climbing can be used to solve problems that have many solutions but where some solutions are better than others. The algorithm is started with a random solution. It sequentially makes small changes to the solution, each time improving toward a better solution. At some instance, the algorithm arrives at a point where it cannot see any improvement anymore, at which point the algorithm terminates. Furthermore, it is not guaranteed that hill climbing will always find the global optimum point (Burke & Kendall, 2005).

2.6.2.2 Simulated Annealing

Simulated Annealing (SA) is a generic probabilistic meta-algorithm for the global optimisation problem, namely locating a good approximation to the global optimum of a given function in a large search space. Simulated annealing is a general method for making the escape possible from local minima by allowing jumps to higher energy states. Since its introduction by Kirkpatrick et al., SA has gained popularity in solving hard combinatorial problems (Kirkpatrick, Gelatt, & Vecchi, 1983).

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. SA is a stochastic search approach capable of escaping local optima by using a transition probability. The transition probability depends on two factors: (1) the difference between the objective values of current solution and the candidate solution, and (2) a parameter known as temperature.

At each step, the SA considers some neighbour S′ of the current state S, and probabilistically decides between moving the system to state S′ or remain unchanged in state S. The probabilities are chosen so that the system ultimately tends to move to states of lower energy. Typically this step is repeated until the system reaches a state that is good enough for the application or until a given computation budget has been exhausted. The SA flow chart is given in Figure 2.3.

Simulated Annealing is a problem independent stochastic optimisation algorithm. If 'data structure of the solution' and 'neighbourhood structure' can be defined efficiently then it is a very effective tool in solving combinatorial optimisation problems optimally. Therefore, it is chosen as the main optimisation algorithm in WITNESS optimizer.

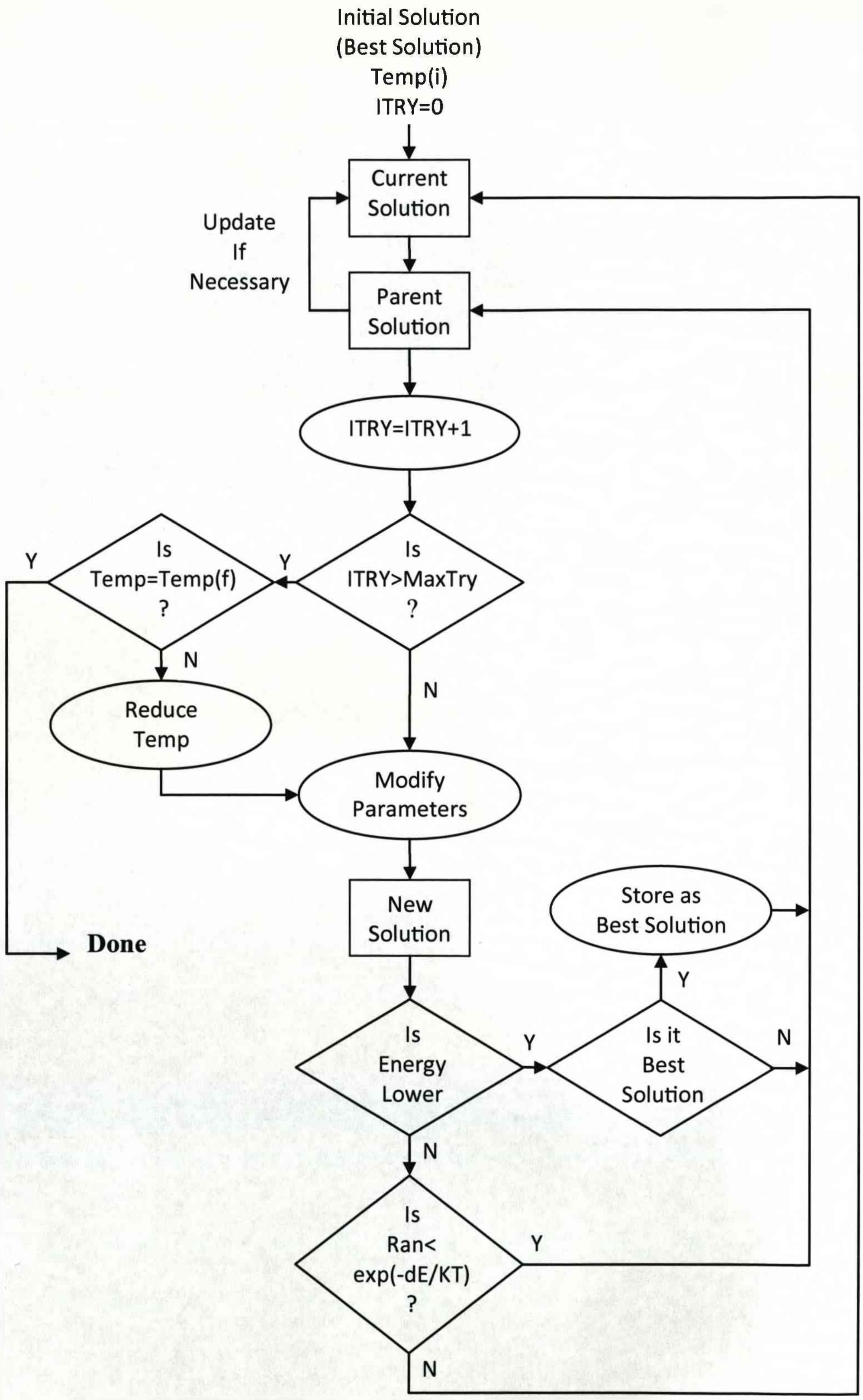


Figure 2.3: Simulated Annealing Flow Chart (Luke, 2003).

Sridhar and Rajendran described three perturbation schemes to generate new sequences for solving the scheduling problem in cellular manufacturing system (Sridhar & Rajendran, 1993). Suresh and Sahu have used SA for assembly line balancing. They only considered single objective problems and found that SA performed at least as well as the other approaches (Suresh & Sahu, 1994). Meller and Bozer have applied SA to facility layout problems with single and multiple floors. The facility layout problem is highly combinatorial in nature and generally exhibits many local minima. SA achieves low-cost solutions that are much less dependent on the initial layout than other approaches (Meller & Bozer, 1996).

Baykasoglu and Gindy developed a SA-based procedure for dynamic layout problem. A Simulated Annealing algorithm with a simple but effective data structure and neighbourhood generation mechanism is proposed and the results outperformed other solutions in literature for this complex optimisation problem (Baykasoglu & Gindy, 2001). A multi-period multi-stop transportation planning problem in a one-warehouse multi-retailer distribution system was studied to determine the routes of vehicles and delivery quantities for each retailer (Kim et al., 2002). They have suggested a two-stage heuristic algorithm based on SA as an alternative for large problems that cannot be solved by the column generation algorithm in a reasonable computation time to minimize the total transportation distance for product delivery over the planning horizon while satisfying demands of the retailers. The efficiency of SA in solving combinatorial optimisation problems is very well known.

Chapter 3
Modelling and Simulation

3.1 Introduction

Simulation Models are typically a part of a case study commissioned by manufacturing management to address a particular set of problems. Simulation can reduce the cost of analysing a problem in real-time in the real-world situation. The models created are tuned in order to represent the real-world situation with an appropriate level of fidelity in order to find the desirable solution. As a theoretical investigation, the model for this research is not based on a particular manufacturing system. A genetic manufacturing system is adopted for this research. In this chapter, a modular design of a manufacturing system together with the objectives, input/output data and data collection criteria are presented.

3.2 Simulation Software

The increasing need for industry to improve manufacturing practice has provided the stimulus for the creation of new technologies and methodologies. Manufacturing systems need to be designed to serve a more demanding and fickle market than ever before. Determining or predicting the operational performance of an existing or reorganized manufacturing system is a challenge. One efficient way to do this is to construct a model using computer-based simulation.

The innovation of simulation language or simulator reduced the threshold of using simulation. It has reduced the amount of computer time and increased the intervention between simulation and decision making in a great degree.

3.2.1 An overview of WITNESS

WITNESS is a premier visual interactive simulation software produced by Lanner Group Ltd. It is the culmination of more than two decades' experience with computer-based simulation. It is capable of modelling both discrete and continuous systems and uses a standard Windows interface. WITNESS has served the simulation market for more than ten years and engaged over 1,000 customer projects.

More and more simulation users are benefiting from the development of the packages every year. The company is leading the way in delivering operational applications with specific industry and public sector practice 'built in' (Lanner Group 2008). The WITNESS simulation package was developed specifically to model manufacturing systems. It is a hybrid of a simulator and a general simulation language. The interface provides the user with a set of menu options, which can be used to develop a generic model of a manufacturing system. The package can use on both standalone personal computers and laptops.

WITNESS simulation model imitates a target system using elements. The most commonly used discrete elements are 'parts', 'buffers' and 'machines'. They are displayed as dynamic icons and represent tangible entities in real-life situation. Logical elements represent control and information aspects of the model which include attributes, variables, distributions and functions. Parts are transferred between elements according to input and output rules which represent different decisions.

A WITNESS model can be run immediately after the elements are defined, displayed and detailed. A logical link can be established at this stage. Step-by-step simulation models can be accomplished to simulate comprehensive manufacturing system models.

Statistical reports are generated automatically after simulation runs. The reports which cover all elements (Cf. Section 3.3.1) in the systems can be used to help to develop the model or aid system analysis.

3.3 Why WITNESS

WITNESS is one of the leading software products in the field of visual interactive simulation. It was chosen for this research because of the following key features:

- Simple and powerful building block design
- Modular and hierarchical structure
- Easy to use standard Windows PC implementation
- Extremely interactive
- Powerful range of logic and control options

- Elements for discrete manufacture
- Comprehensive statistical input and reports
- Quality graphical displays
- Linkage-databases (ORACLE, SQL Server, Access, etc), direct spreadsheet links in/out, XML save formats, HTML reports, links from partner BPM and CAD applications

WITNESS is available in two versions: the Manufacturing Performance Edition and the Service and Process Performance Edition. The Manufacturing Performance Edition of WITNESS contains the core WITNESS ability to model any type of simulation problem whether it is production, logistical or service based. The terminology used is tailored to manufacturing as this edition of WITNESS is targeted primarily at organisations with manufacturing industry bias.

WITNESS is widely used and well proven. In simulation software comparison studies WITNESS regularly was proven to perform well, for instance, WITNESS possess more desirable features than other simulators (Banks, Aviles, Mclaughlin, & Yuan, 1991); its user-friendliness visual and coding aspects were rated highly due to the testability and input/output features (Hlupic & Paul, 1995). The latest survey on Simulation Software based on a questionnaire answered by 65 software package vendors can be found on *OR/MS Today* (Swain, 2005). The questionnaire covers the aspect of software applications, primary market, computer system requirements, model building details as well as the support and training information supplied with the software packages. Overall, WITNESS was selected due to its suitability for modelling and simulating of a detailed manufacturing system.

3.4 WITNESS model structure

The basic steps or the procedure required when building a simulation model are shown in Figure 3.1. Problem formulation and data collection are the utmost important steps of a modelling process as they will determine the quality of the model. The most time-consuming step amongst them is model verification.

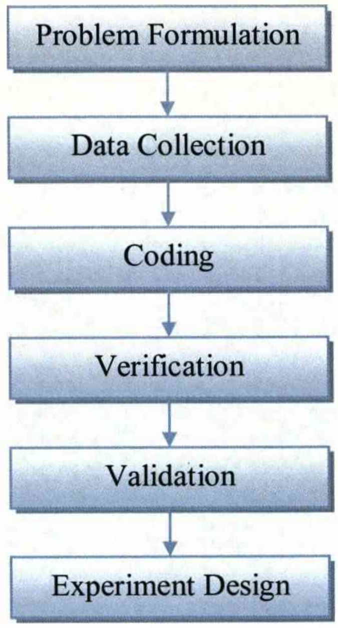


Figure 3.1: Simulation model development process.

3.4.1 Modelling elements

The WITNESS simulation package is capable of modelling a variety of discrete and continuous elements. Each element is in a *state* such as idle, busy, blocked, broken-down and waiting labour depending on the type of element.

As mentioned above, this research deals with a discrete event model hence the basic elements in the model are discrete elements such as Parts, Buffers, Machines and Labours as shown in Figure 3.2. These elements can be defined, detailed and then displayed in the model layout windows.

Parts are objects that travel from one location to another. In manufacturing, Parts normally refer to material or sub product in the system. Parts flow can be made by pull/push control logic or part routing settings.

Buffers are passive storage areas of finite capacity. A part can be optionally ejected from a buffer under certain condition or simply waiting for pulling out from other elements such as machines. Combinations of First-In-First-Out / Last-In-First-Out dispatching rules are possible, as well as front and rear location of the buffer.

Machines are the functional blocks of WITNESS that drive the simulation. A variety of machine types can be simulated, such as single, batch, assembly, production and so on. Setup, breakdown, shift and costing can be defined in the machine detail dialogue, which is very useful for modelling real-life scenarios.

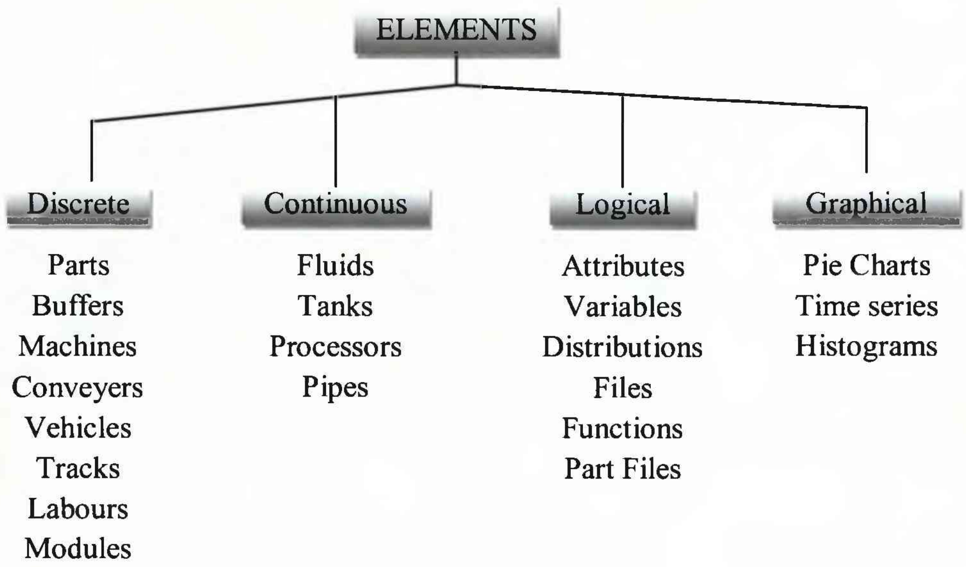


Figure 3.2: WITNESS elements and element groups.

Labour can be used to model both human and physical resources (for example, tools, people or equipment) which may be required by other elements for processing, setting up, repair, cleaning and so on. Labour can be attached to a machine element or a part element. It is fairly straightforward to model operators and manage human resources by naming operators and applying shifts to operators.

3.4.2 Modelling logic

In the WITNESS model development process, it is important to understand the basic control logic involved in the detailing stage at which modellers insert element details. Running a model means the initiation of a sequence of events according to simulation time. For instance, a part entering into the system is an event, and the part being stored into a buffer is another event followed by the part entering into a machine waiting for process. A part being processed in a machine involves certain kind of events, decisions or activities. Understanding the order of execution for every element is the key fact of successful modelling. The order of execution for a machine element is shown in Figure 3.3.

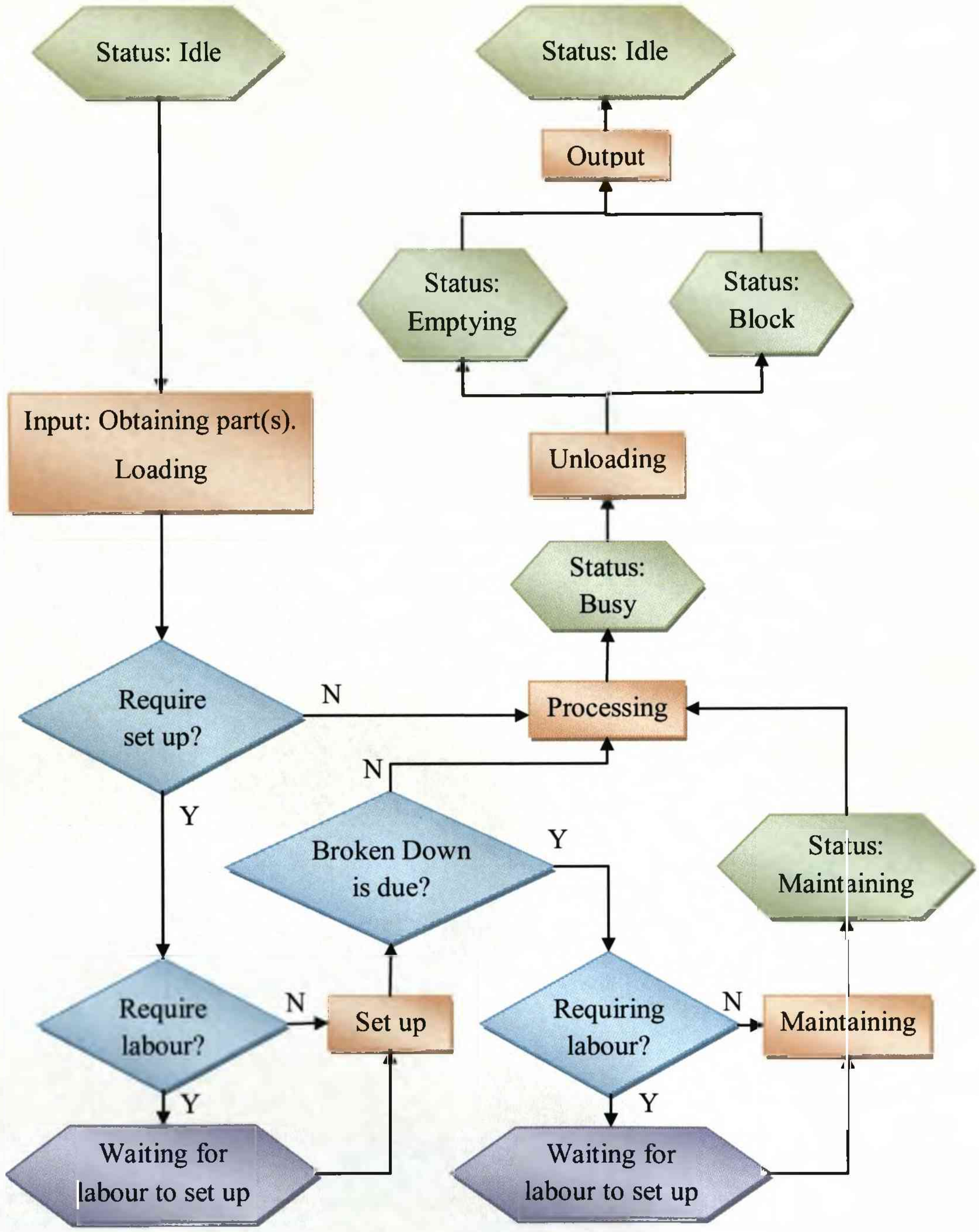


Figure 3.3: Order of execution for a machine's cycle.

A machine is triggered into an activity when a request for a part(s) is successful. The status (green block in Figure 3.3) is recorded automatically when the process develops into different stages. These stages (orange block in Figure 3.3) include *load, set-up, breakdown, maintain, process and unload*. A typical machining process involves at least three activities, i.e., *loading, processing and unloading*. The attribute of these three actions must be defined when a machine is created. Even though it is unlikely to have zero set-up and breakdown, it is usual not to model them in order to simplify the model.

Other elements such as buffers, conveyers have their own execution logic. The machine's activity logic is the most complicated and important. In WITNESS, buffer elements do not pull parts in or push them out. Hence, parts flow through buffers use an active element such as a machine, conveyor, pipe etc.

Generally speaking, all physical entities of the manufacturing system must be defined and detailed in the early stages of modelling process. As a Windows simulation application, WITNESS provides different windows to assist the defining process for each element. Machine elements can be assigned with values to describe their characteristics such as the name of machine, type of machine, cycle time, input output rules, labour rules, set up time, breakdown scenarios as given in a machine detail windows shown in Figure 3.4.

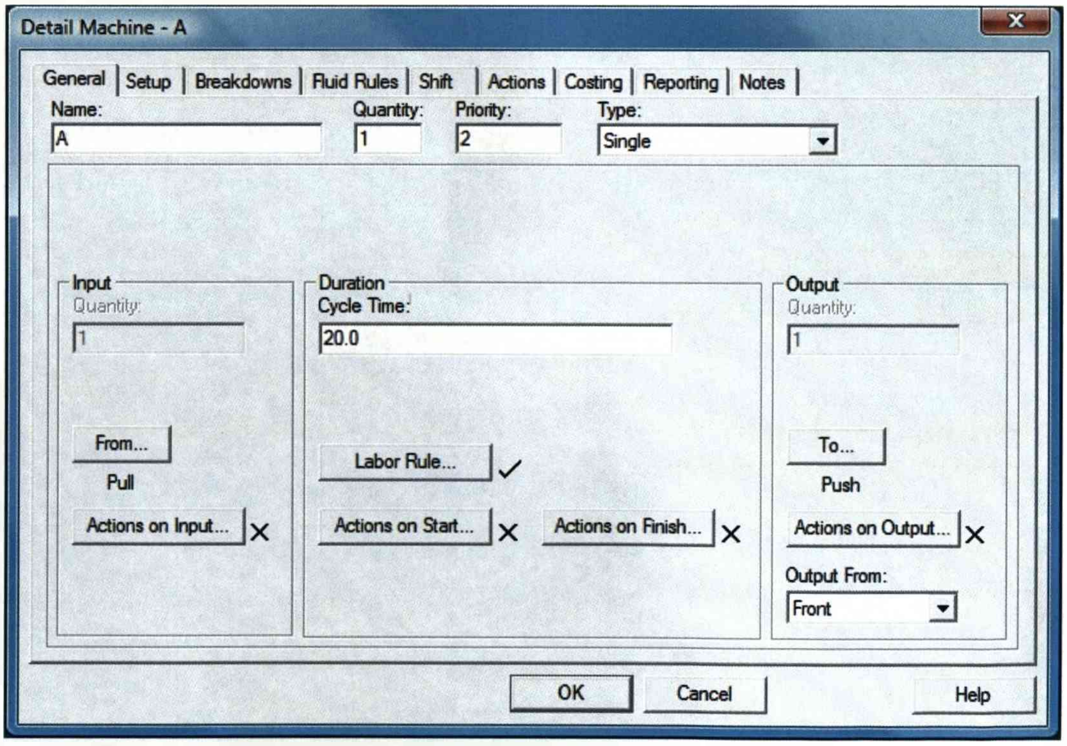


Figure 3.4: Machine element detailing window.

This modelling technique makes it very straightforward and quicker to build a model from scratch. Nevertheless, it limits the freedom of the modeller. It is not always flexible enough to represent a scenario as needed. For instance, there is only one place to define the machine cycle time. Machines often need to process different parts with different cycle times. Introducing few parameters and connecting them with specific parts are necessary. This added complexity is unavoidable during the WITNESS modelling process. Finding the shortest way to tackle the problems and reducing the added complexity is a criterion when comparing different models.

3.4.3 Materials flow

After defining, displaying and detailing the elements of a model, the model can be run immediately. Materials flow through the manufacturing system can be observed on the screen. Basic activities include parts being pulled into the system, stored in a buffer, pulled onto a machine and pushed or shipped out once the process is completed. Materials handling is omitted in most models unless one of the objectives of the model is to study the materials handling process. *Pull* and *Push* statements are the engine of the WITNESS materials flow. For example, a *pull* statement '*Pull from buffer01*' is used in the input request of a machine element to get parts. The new version WITNESS 2007 added visual *pull* and *push* button to assist the modelling process. A *pull* and *push* statement can be done by clicking icons of elements rather than input the actual codes. This improvement reduces the modelling time effectively.

It is quite common that parts are pulled from alternate locations at the same time for an assembly process (Aytuğ & Doğ an, 1998). Apart from *pull* and *push* statement, *sequence* statement is very useful for the input of assembly machines and multi parts multi machines scenarios. In addition, three built-in options *sequence/Wait, sequence/Next* and *sequence/Reset* give options when it fails to input from or output to an element or location. *Sequence/Wait* option is the most common production control logic where a machine waits until it can input or output a part. An example statement is given as follows:

SEQUENCE /Wait Part1 out of Buffer1#(1), Part2 out of Buffer2#(1), Part3 out of Buffer3#(1)

The above command statement directs a machine to pull three different parts from Buffer1, Buffer2 and Buffer3 and assemble them together. Replacing the *Wait* by *Next,* if machine fails to use the element, it attempts to use the one after and so on until there is an available element to use. The *Reset* option will reset the sequence and go back to the first element in the list.

Output rules (e.g. PUSH, PERCENT and SEQUENCE) are relatively easier. All machines can use *Push to Next-Destination* to specify the dispatching location.

After turned on the "element flow" button, all the materials flow sequence can be shown in the system layout window. In this way, it is almost impossible to have wrong codes for materials flow sequence.

3.5 Classic Two Products model

In order to gain experience in designing, creating and running a simulated manufacturing model, a small manufacturing model was created first. This exercise served as an essential part to become familiar with the simulation procedures.

3.5.1 Problem description

The example is taken from Goldratt (Goldratt, 1990). The case can be described as two product four machine problem (2P/4M) since it is a process which makes just two products P, Q and uses four machines A, B, C and D. The manufacturing environment is represented in Figure 3.5. The objective of the exercise is to make a decision as to how to best maximize the profit of this process.

There are three raw materials (RM) and one purchased part which are used to make the two end products P and Q. One unit each of RM1 and RM2 combined with one purchased part constitutes the material requirement for product P. One unit each of RM2 and RM3 is assembled to form one product Q. There are A, B, C and D four machines where A, B and C are for materials processing and D is for the final assembly. As there is no multitasking involved, one machine can only process one job at a time.

In addition, the operating expense for the whole process is £6000 per week. In this ideal situation, the availability of all machines is 60 minutes per hour, 8 hours per day, 5 days per week, i.e. 2400 minutes per week. There is no waiting, setting up or machine break down.

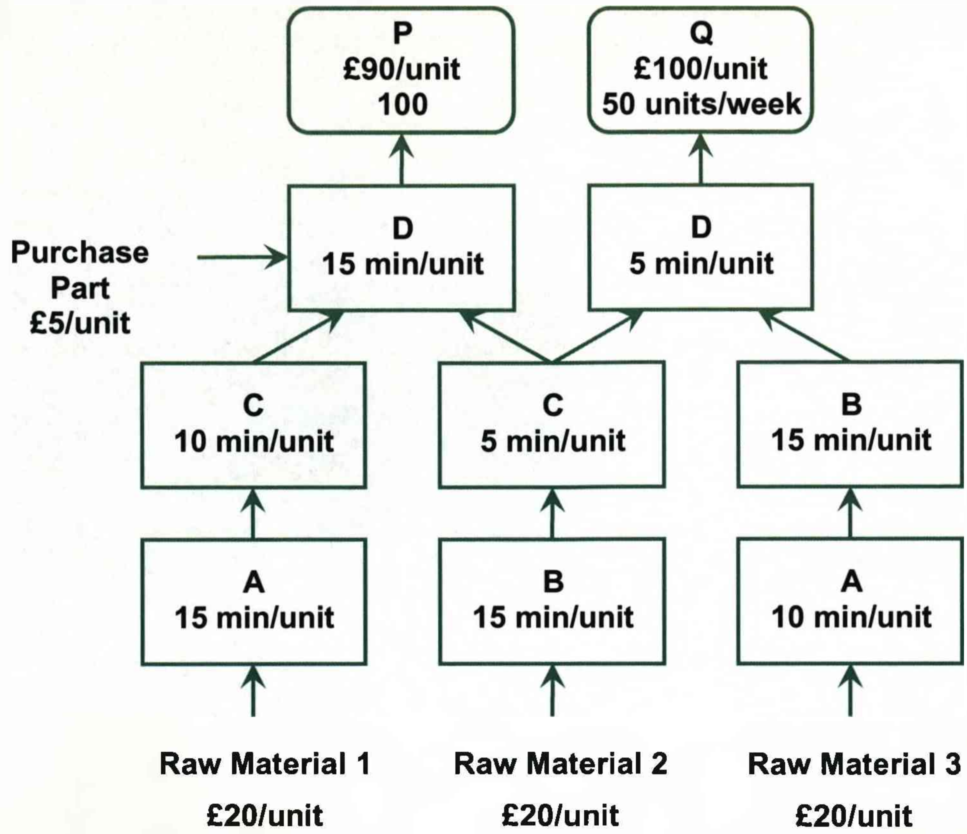


Figure 3.5: 2P/4M problem demonstration.

3.5.2 Objectives of the model

Based on the above information a simulation model is built to analyse the proposed manufacturing system. The aim of building this model is to answer the following questions:

1. Can the simulation model find the bottleneck machine?
2. How does the system react to varying demand or supplier delays?
3. How much extra capacity is required for bottleneck machine reconfiguration?
4. What happens when the system has more than one bottleneck machines?
5. How to simulate the impact of scheduling on output?
6. What is the cost of reconfiguration?
7. What is the effect of selling price change?
8. What is the outcome of machine breakdown?

3.5.3 Layout of the model

The physical layout of the system is not an issue in this case study but the machine performance and system scheduling. Therefore, the simulation model is built according to the processing jobs. Machine icon in the model is a visual machine processing job rather than a physical machine. In other words, the number of machine icons present in the simulation model layout signifies the number of jobs that exist in the system.

Figure 3.6 shows the layout window of the simulation model and Figure 3.7 shows the actual physical layout of the manufacturing system. Blue lines connect all the elements in the system and illustrate the element flow and processing sequence. The benefit of using such a technique to model the system is obvious between two layout windows.

The benefits of visualisation of the system are for the consideration of the following questions: How many processes are there for each part before the assembly? What is the machine routing for each part? How many parts are processed in one machine? How many products are there in the system? All these questions can be read from the Figure 3.6 easily. It is also possible to allocate machine cycle time individually for each parts with this method. It offers the benefit of easy control and an overview of machine-part processing routing. The benefits that have been discussed have the effect of streamlining the process of creating the reconfiguration simulation experiments, allowing a greater complexity of simulation to be created in a shorter period of time.

In this model, machine A → A1/A2 never work simultaneously, so as machine B → B1/B2, machine C → C1/C2, and machine D → D1/D2. Hence they act as one machine and the performance data at the combination of two machines is identical with simulating just one machine. The simulation model does not need to match the real life system as long as the data the model generated is not affected. This is a general rule to simplify a simulation model. As a matter of fact, no simulation model will be 100% real-life matching, it is unavoidable to simplify the model and filter the unnecessary data out.

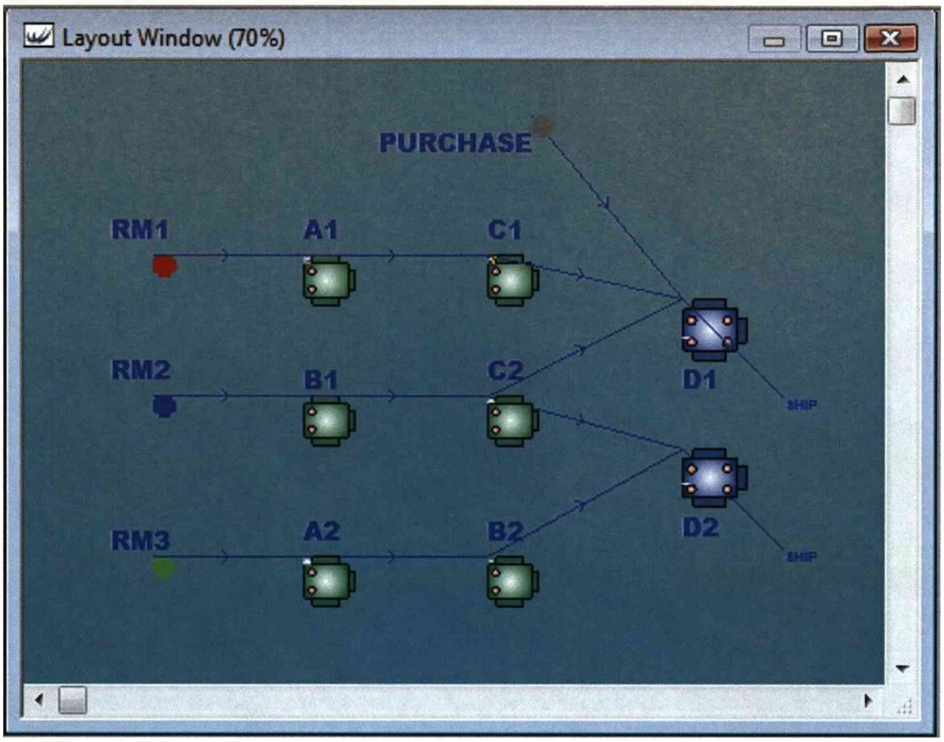


Figure 3.6: Layout window of the simulation model.

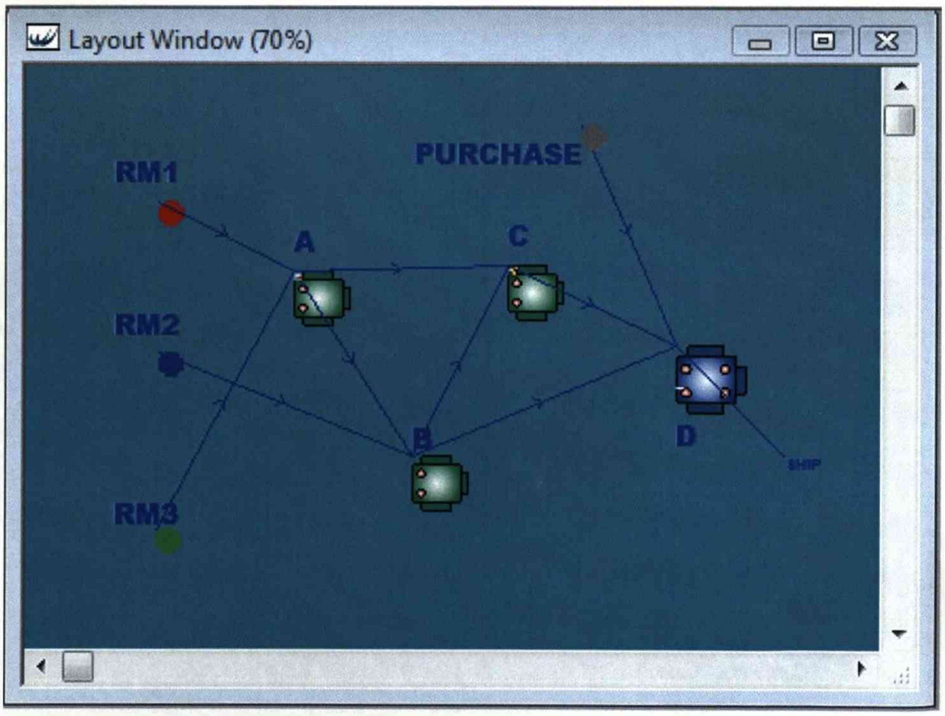


Figure 3.7: Physical layout of the manufacturing system.

3.5.4 Eight Scenarios

After verification and validation of the model, eight simulation tests were designed to study the system and machine performance. The descriptions of these eight scenarios are as follows:

Table 3.1: Eight scenarios of the 2P/4M problem.

Scenario	Descriptions	Equation statements
1	All machines are under full capacity to meet market requirement. Machine utilisation ≤100%	$T_{a1}\times D_p+T_{a3}\times D_q\leq M_{Ac}$ $T_{b2}\times D_p+(T_{b2}+T_{b3})\times D_q\leq M_{Bc}$ $(T_{c1}+T_{c2})\times D_p+T_{c2}\times D_q\leq M_{Cc}$ $T_{dp}\times D_p+T_{dq}\times D_q\leq M_{Dc}$
2	The system meets partial market demand. There is one bottleneck machine (Machine B). Machine B's requirement >100% machine capacity.	$T_{a1}\times D_p+T_{a3}\times D_q\leq M_{Ac}$ $T_{b2}\times D_p+(T_{b2}+T_{b3})\times D_q\geq M_{Bc}$ $(T_{c1}+T_{c2})\times D_p+T_{c2}\times D_q\leq M_{Cc}$ $T_{dp}\times D_p+T_{dq}\times D_q\leq M_{Dc}$
3	The system configuration only meets partial market demand. There are two bottleneck machines (machine B and machine A). Machines A and B have the same capacity requirement. Originally machine B's capacity requirement is 125%. In this scenario, both A and B have the same capacity requirement 125%.	$T_{a1}\times D_p+T_{a3}\times D_q\geq M_{Ac}$ $T_{b2}\times D_p+(T_{b2}+T_{b3})\times D_q\geq M_{Bc}$ $(T_{c1}+T_{c2})\times D_p+T_{c2}\times D_q\leq M_{Cc}$ $T_{dp}\times D_p+T_{dq}\times D_q\leq M_{Dc}$
4	The system meets partial market demand. There are two bottleneck machines (machine B and A). Machines A and B's requirement >100% machine actual capacity. Machines A and B have different capacity requirement. A's capacity requirement is constant at 125%. Machine B's capacity requirement changes from 100% to 200%.	$T_{a1}\times D_p+T_{a3}\times D_q\geq M_{Ac}$ $T_{b2}\times D_p+(T_{b2}+T_{b3})\times D_q\geq M_{Bc}$ $(T_{c1}+T_{c2})\times D_p+T_{c2}\times D_q\leq M_{Cc}$ $T_{dp}\times D_p+T_{dq}\times D_q\leq M_{Dc}$
5	Short supply of raw materials. Four situations are analysed. Situation one has limited raw material1; situation two has limited raw material2; situation three has limited raw material3; situation four has limited purchase material.	1. $RM1<D_p$ 2. $RM2<D_p+D_q$ 3. $RM3<D_q$ 4. $P4<D_p$

6	Market demand changes. In this scenario, two situations are analysed. One is both products P and Q market demand increase, the other is both of them decrease.	1. $D_p\uparrow$ $D_q\uparrow$ 2. $D_p\downarrow$ $D_q\downarrow$
7	Price Change	P_p,P_q,P_1,P_2,P_3,P_4 various
8	Machine Breakdown	

Where,

T_{xy} = cycle time (minutes) on machine x for raw material y.

D_p = market demand of product P, D_q market demand of product Q.

M_{Ac} ,M_{Bc} ,M_{Cc} ,M_{Dc} = capacity of machine A,B,C,D respectively.

RM1, RM2, RM3, P4 = available quantity of raw materials and purchased parts.

P_p, P_q, P_1, P_2, P_3, P_4 = the prices for Product P,Q, RM1 to RM3 and purchased part P4 respectively.

3.5.5 Results and conclusions

The 2P/4M problem is a small size manufacturing scenario; hence it can be solved by Linear Programming as well. The simulation results answered all the given questions as listed in Section 3.5.2. Simulation results matched the manual calculation data using Linear Programming and in most scenarios the simulation model gave results more quickly and more comprehensively. Many system performance problems can be read on the simulation window directly. It also proved that the model building technique worked well. With this perpetrating pilot study, a more complex medium size manufacturing model was built for the main reconfiguration investigation.

3.6 Five products and twelve machines model (5P/12M model)

3.6.1 System description

The model is built as a medium size manufacturing processing and assembly system. The elements in the simulation model include machines, raw material parts, buffers, labours and products. There are twelve types of machines in the workshop; amongst them, nine are single processing machines (Machines A, B, C, D, E, F, G, H and I) and three are assembly machines (Machines J, K and L).

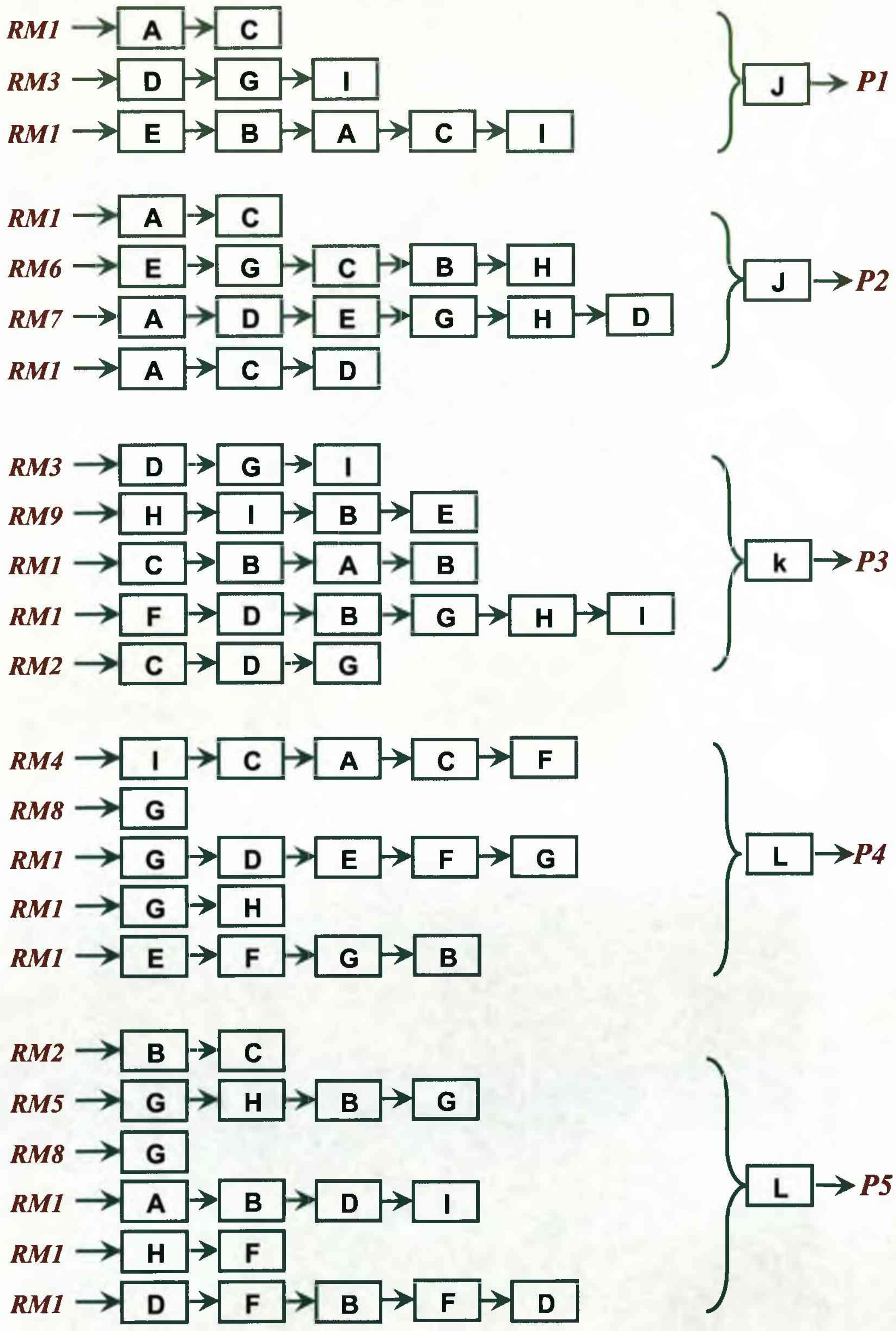


Figure 3.8: 5P/12M processing routing.

Among the twelve types of machines, machines G and H are duplicated in the system due to high capacity requirement while there is only one of each for all other machines. Hence, there are fourteen actual machines in the system. For job processing on Machines G and H, two jobs can be processed simultaneously as there are two machines available. Twenty raw materials (RM1 to RM20) are processed on requisite machines first and then wait for assembly. Processing on each material varies from one step only to as many as six steps. As shown in Figure 3.8, the model produces five final products (P1 to P5) which are assembled with three or more processed materials. Between the five products, RM1, RM3 and RM8 are common parts used on more than one product.

Although this is not a large mass production manufacturing system, the complexity of the machines-parts processing is beyond analytical calculation.

3.6.2 Objectives and requirements of the simulation model

The five products twelve machines model (5P/12M) is built to illustrate the use of a simulation model to support decision-making for manufacturing system reconfiguration. With the understanding of the use of the simulation model the requirements of the simulation model can be stated as follows:

- ✓ Animation of a medium size medium level complex manufacturing system.
- ✓ The model and the machines can be reconfigured.
- ✓ Data input can be varied accordingly.
- ✓ Ability to generate machine performance data.
- ✓ Built-in cost function to assist the analysis on reconfiguration costs.

3.6.3 Assumptions

All simulation models are purpose built for solving one or a few problems for a given system. It is impractical and impossible to simulate a manufacturing system in every detail. In this model, the information is generated according to given objectives. Only the necessary information and data are built into the model. Therefore, a number of assumptions have to be made before building the model as follows:

- ➢ Raw materials are always available.
- ➢ Each machine can only process one job at a time.

- There is no waiting in any machines.
- Set-up, maintenance and breakdown are not included in the simulation model.
- It is assumed that transportation and materials handling is automated and takes zero time.
- Operators are always available unless already employed on other jobs. No shift pattern is set up for operators.
- Operators are allocated to specific machines. One machine cannot use an operator from another machine.
- There is no pre-exemption on operators, which means once the process job has started it will have to finish the entire job.
- All buffers and warehouse have unlimited storage capacity.
- There are no defects on products.
- There are no alternative routes and no products other than the five listed are produced in the system.
- The system only produces enough to satisfy market demand, there is no over production or preproduction.

3.6.4 Data input

The data used to build the model are generated randomly within a reasonable range. Data can be inputted using element detail dialogue window directly from WTINESS. For experimental purpose, some variable data can be formed as a Text document or Excel file outside WITNESS. Variables are logical element in WITNESS model which can be defined and detailed in element detail dialogue window as well. Variable can be controlled by WTINESS Scenario Manager, WITNESS Optimizer or simply WITNESS reads a value from a Text or Excel file.

For example, variable T_{a1} is the cycle time of machine A to process raw material 1. It appears in Machine A's detail dialogue window. Before running the model, the system will read T_{a1} from a file call "cycle_time" to give T_{a1} a real value. In this model, there are three input files.

Cycle_time: machine_part cycle time matrix
Price: the price of raw materials and products
Demand: Demand quantity of the five products per quarter

It is practical to increase the flexibility and modifyability by using variables to fill up element detail dialogue windows. Figure 3.9 illustrates the benefits of this approach.

Figure 3.9: The advantage of isolating input data.

Input data files can relate directly to the output results. A series of scenario analysis can be carried out without changing anything inside the WITNESS model.

3.6.5 Layout of the model

Following up the Goldratt's 2P/4M manufacturing model, the technique used to arrange the machine processing job is applied in this model as well. In Figure 3.10, it is obvious that the advantage of the modelling technique benefits more in a larger model. It reduces the complexity on modelling a system significantly and helps users to understand the processing order in the system by visual inspection of the model.

The simulation layout window shows the operation process of the manufacturing system given in Figure 3.10. Twenty raw materials with red dot icon are pulled into the system and processed by one or few machines (green machine icon) and then assembled into five final products by the blue assembly machines. All machines have buffers in front of them to store the parts waiting for processing. There are twenty buffers for all twenty semi-finished parts before the assembly. All machines processing jobs require the presence of a matching operator. Therefore, there are twelve types of labour elements (Operator A ~ Operator L). There are two Operators G and two Operators H since there are two Machine G and Machine H.

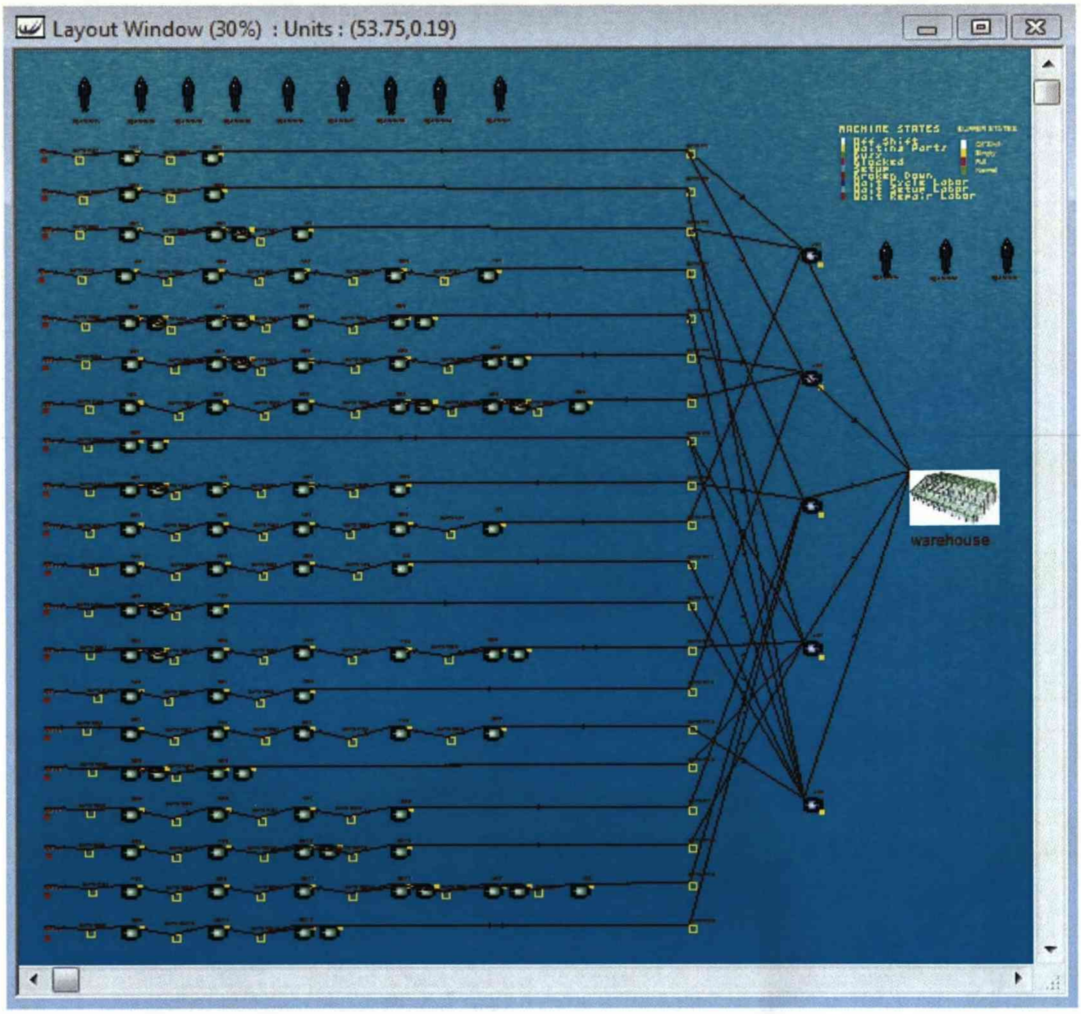


Figure 3.10: Model layout window.

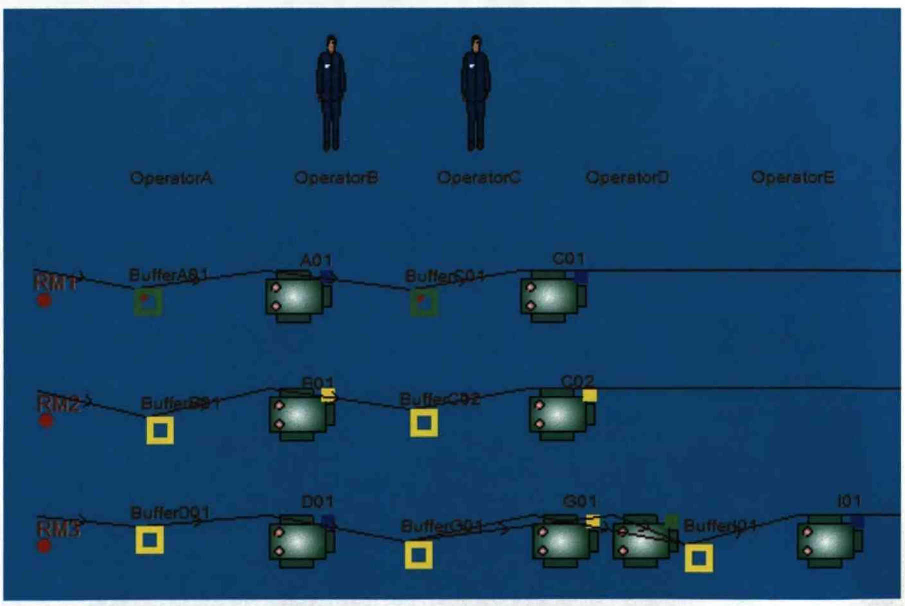


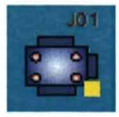
Figure 3.11: Enlarged screenshot of the layout window.

Where the icons stands for:

Raw materials

Buffers

Single processing machine



Assembly machine

Operators

Warehouse

Figure 3.11 is an enlarged screenshot on the top left corner of the whole model layout window. As different to Figure 3.10, the model is in the running mode. The colours of machine and buffer icons represent the current states of the elements. With the help of machine buffer states key as illustrated in Figure 3.12, it is obvious that one of the Machine G is processing RM3 since the icon is green. BufferA01 and BufferC01 are in use as the colour of the buffer icon is green and the red dots inside represent raw materials.

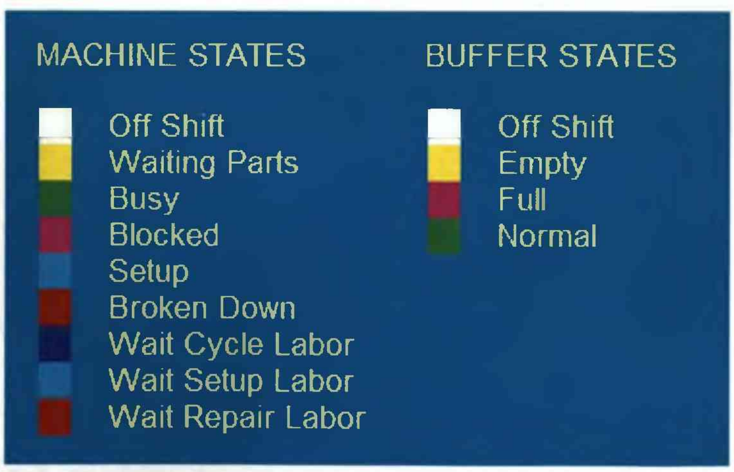


Figure 3.12: Machine buffer states key.

The machine state icons in Figure 3.11 show that there are four machines in the state of waiting for operators. The reason that there are so many machine icons in this state is due to the arrangement used in building the model. One actual machine uses multiple machine icons in the model to present the different processing jobs. However, they still have to act as one physical machine. Thus, there is only one machine element in the model that can be busy at a time. As one operator is shared between many machine elements under one physical machine, it makes it possible to control the working sequence and animate the physical machine with several machine elements of the same machine in the simulation model. For this reason, there are many machine elements in the waiting operator states. It can also be understood as a particular part is waiting to be processing on a particular machine after it is free from processing other parts. For example, Machine A has seven different processing jobs (A01, A02 ... A07). As operator A is on machine A02, all the other six jobs will have to wait until the operator is available. Here, the use of operator is a very practical way to ensure that the simulation result is validated and match the physical manufacturing system.

The machine and buffer icon is yellow before running the simulation. When the machine or buffer icon is yellow during the simulation run, it means that the machine is idle or the buffer is empty. The operators' icons will move to the machine when it is in the processing state to indicate the availability of the operator. In Figure 3.11, operators B and C are not in operation state hence their icons are still in the waiting area. It also indicates that machines B and C are idle while other machines are busy.

All the data used to build the model are included in Table 3.2 which is arranged in the process sequences. The data includes machine cycle-time, raw materials costs, sale price of the five products and market demand of these products. With the assumptions made in Section 3.5.3, only such data is needed to build the simulation model and generate results for system performance analysis.

Table 3.2: System process data.

RM1	A01	C01				
£5	3mins	6mins				
RM2	B01	C02				
£20	6mins	6mins				
RM3	D01	G01	I01			
£15	3mins	8mins	8mins			
RM4	I02	C03	A02	C04	F01	
£20	8mins	7mins	3mins	7mins	12mins	
RM5	G02	H01	B02	G03		
£10	8mins	17mins	5mins	9mins		
RM6	E01	G04	C05	B03	H02	
£15	10mins	8mins	6mins	6mins	18mins	
RM7	A03	D02	E02	G05	H03	D03
£5	4mins	4mins	11mins	10mins	17mins	3mins
RM8	G06					
£10	10mins					
RM9	H04	I03	B04	E03		
£20	18mins	10mins	5mins	11mins		
RM10	E04	B05	A04	C06	I04	
£15	10mins	6mins	4mins	6mins	10mins	
RM11	A05	B06	D04	I05		
£20	3mins	5mins	4mins	8mins		
RM12	H05	F02				
£5	18mins	12mins				
RM13	G07	D05	E05	F03	G08	
£5	8mins	3mins	11mins	13mins	10mins	
RM14	A06	C07	D06			
£20	5mins	7mins	3mins			
RM15	D07	F04	B07	F05	D08	
£10	4mins	12mins	5mins	12mins	4mins	
RM16	G09	H06				
£5	10mins	15mins				
RM17	C08	B08	A07	B09		
£10	7mins	5mins	3mins	5mins		
RM18	E06	F06	G10	B10		
£15	11mins	12mins	8mins	5mins		
RM19	F07	D09	B11	G11	H07	I06
£20	12mins	4mins	5mins	8mins	15mins	8mins
RM20	C09	D10	G12			
£5	6mins	4mins	8mins			

RM1	J01	P1
RM3	10mins	£200
RM10		55/WK
RM1	J02	P2
RM6	12mins	£400
RM7		40/WK
RM14		
RM3	K	P3
RM9	20mins	£500
RM17		25/WK
RM19		
RM20		
RM4	L01	P4
RM8	15mins	£600
RM13		30/WK
RM16		
RM18		
RM2	L02	P5
RM5	17mins	£750
RM8		25/WK
RM11		
RM12		
RM15		

Where:

The price under red RM is the purchase price of raw materials. Time under single green machine and blue assembly machine is the cycle time. Data under five purple products is the sale price and weekly market demand.

3.6.6 Logical control

Manufacturing companies run their workshop on a day to day basis. Behind the obvious routing there are rules for organising and controlling the whole system to make sure it runs smoothly. In simulation models, these rules are changed into computer code and presented as modelling logic.

(A) The arrival of raw material

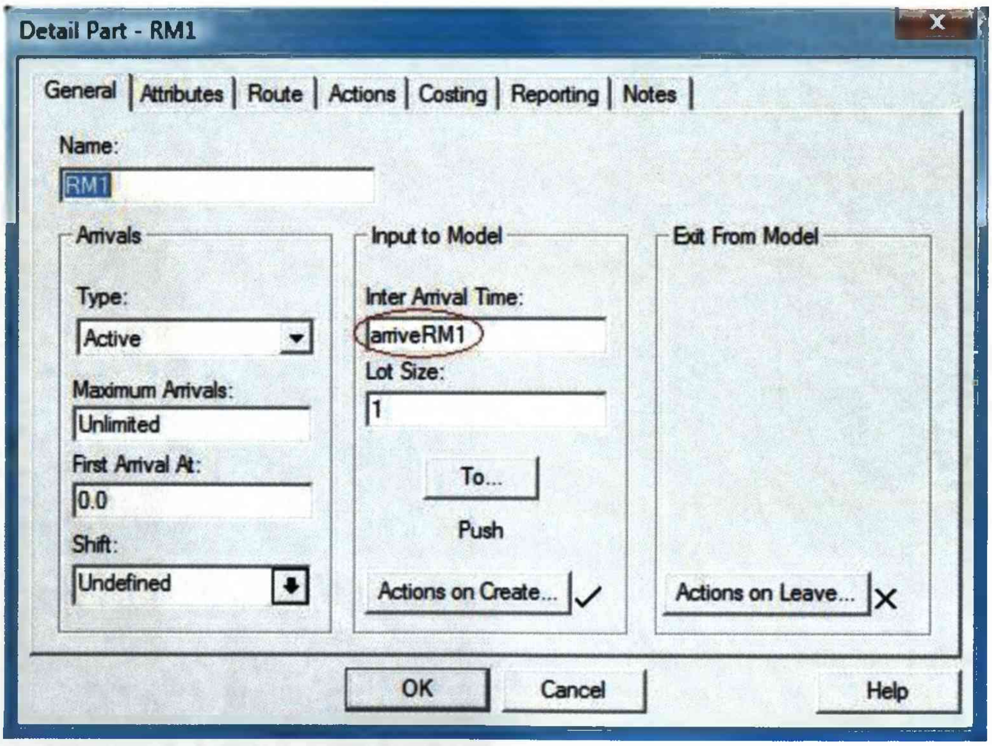


Figure 3.13: Parts detail window.

As shown in Figure 3.13, a detailed dialogue window is used to provide information for part elements in WITNESS models. The part detail window for Raw Material 1 gives information for RM1 such as maximum arrivals, first arrival time, inter arrival time and lot size after creating an active part. If there are logical mistakes with the information, a pop-up error message will point out the mistakes when the model is prompted to run. According to the assumptions, raw materials are always available with unlimited supply.

In the *Inter Arrival Time* field of the detail dialogue, the amount of time that elapsed between the arrivals of each part is stated. The WITNESS part file element specifies that the values here are absolute, which means that they refer to specific times in the simulation run. Without this information, the model will not run. The inter-arrival time can be a constant value if parts arrive at the same interval each time. In many case when the arrival of parts is uncertain a distribution can be used in the field, such as RANDOM(1,10) or NORMAL(1,10). It is most used in customer services model when customers (parts) arrive in a certain pattern.

In this research, the supply of parts is not an issue. Therefore, the inter-arrival time should be set according to the demand of processing machines. Twenty real variables (arriveRM1 to arriveRM20) are set to represent inter-arrival time for each raw material. They are calculated in the initialise actions of the model.

There are many ways to decide the inter-arrive time. In this case, machine processing is rather complicated. It is not straightforward to determine the best way to calculate the inter-arrive time of each raw material. There is a buffer set up in front of each machine in this model, which prevents machine blockage by over feeding parts. Figures 3.14 and 3.15 are two ways to calculate material inter-arrival time. Figure 3.15 uses the maximum single machine processing time of each single part. For example, RM1 is processed on Machine A for 3 minutes then Machine C for 6 minutes. Choosing the maximum processing time, the inter-arrival time of RM1 is 6 minutes. This method does not consider other parts processing time for a same product. RM1, RM3 and RM10 are three parts required for assembly a product P1, RM10 takes at least 10 minutes to process. Therefore, there is spare time for parts to wait until the completion of other parts before assembly. Figure 3.14 uses the maximum single machine processing time of each product. Common parts RM1, RM3 and RM8 are required for more than one product; therefore they still use processing time on the single part to guarantee the supply of assembly. Hence, with the exception of common parts, all parts for one product will share the same inter arrival time. After testing both methods on the simulation model, the second method is used in further simulation research because it reduces the work in progress level efficiently.

$\mathrm{arriveRM1} = \mathrm{MAX}\ (T_{a1},T_{c1})$

$\mathrm{arriveRM3} = \mathrm{MAX}\ (T_{d1},T_{g1},T_{i1})$

$\mathrm{arriveRM10} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{a1},T_{c1}),\mathrm{MAX}\ (T_{d1},T_{g1},T_{i1}),$

$\mathrm{MAX}\ (T_{e4},T_{b5},T_{a4},T_{c6},T_{i4}))$

$\mathrm{arriveRM6} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{a1},T_{c1}),\mathrm{MAX}\ (T_{e1},T_{g4},T_{c5},T_{b3},T_{h2}),$

$\mathrm{MAX}(T_{a3},T_{d2} + T_{d3},T_{e2},T_{g5},T_{h3}),\ \mathrm{MAX}\ (T_{a6},T_{c7},T_{d6}))$

$\mathrm{arriveRM7} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{a1},T_{c1}),\mathrm{MAX}\ (T_{e1},T_{g4},T_{c5},T_{b3},T_{h2}),$

$\mathrm{MAX}(T_{a3},T_{d2} + T_{d3},T_{e2},T_{g5},T_{h3}),\ \mathrm{MAX}\ (T_{a6},T_{c7},T_{d6}))$

$\mathrm{arriveRM14} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{a1},T_{c1}),\mathrm{MAX}\ (T_{e1},T_{g4},T_{c5},T_{b3},T_{h2}),$

$\mathrm{MAX}(T_{a3},T_{d2} + T_{d3},T_{e2},T_{g5},T_{h3}),\ \mathrm{MAX}\ (T_{a6},T_{c7},T_{d6}))$

$\mathrm{arriveRM9} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{d1},T_{g1},T_{i1}),$

$\mathrm{MAX}\ (T_{h4},T_{i3},T_{b4},T_{e3}),\mathrm{MAX}(T_{c8},T_{b8},T_{a7},T_{b9}),$

$\mathrm{MAX}\ (T_{f7},T_{d9},T_{b11},T_{g11},T_{h7},T_{i6}),\mathrm{MAX}\ (T_{c9},T_{d10},T_{g12}))$

$\mathrm{arriveRM17} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{d1},T_{g1},T_{i1}),$

$\mathrm{MAX}\ (T_{h4},T_{i3},T_{b4},T_{e3}),\mathrm{MAX}(T_{c8},T_{b8},T_{a7},T_{b9}),$

$\mathrm{MAX}\ (T_{f7},T_{d9},T_{b11},T_{g11},T_{h7},T_{i6}),\mathrm{MAX}\ (T_{c9},T_{d10},T_{g12}))$

$\mathrm{arriveRM19} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{d1},T_{g1},T_{i1}),$

$\mathrm{MAX}\ (T_{h4},T_{i3},T_{b4},T_{e3}),\mathrm{MAX}(T_{c8},T_{b8},T_{a7},T_{b9}),$

$\mathrm{MAX}\ (T_{f7},T_{d9},T_{b11},T_{g11},T_{h7},T_{i6}),\mathrm{MAX}\ (T_{c9},T_{d10},T_{g12}))$

$\mathrm{arriveRM20} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{d1},T_{g1},T_{i1}),$

$\mathrm{MAX}\ (T_{h4},T_{i3},T_{b4},T_{e3}),\mathrm{MAX}(T_{c8},T_{b8},T_{a7},T_{b9}),$

$\mathrm{MAX}\ (T_{f7},T_{d9},T_{b11},T_{g11},T_{h7},T_{i6}),\mathrm{MAX}\ (T_{c9},T_{d10},T_{g12}))$

$\mathrm{arriveRM4} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{i2},T_{c3} + T_{c4},T_{a2},T_{f1}),\mathrm{MAX}\ (T_{g6}),$

$\mathrm{MAX}\ (T_{g7} + T_{g8},T_{d5},T_{e5},T_{f3}),\mathrm{MAX}\ (T_{g9},T_{h6}),\mathrm{MAX}\ (T_{e6},T_{f6},T_{g10},T_{b10}))$

$\mathrm{arriveRM8} = \mathrm{MAX}\ (T_{g6})$

$\mathrm{arriveRM13} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{i2},T_{c3} + T_{c4},T_{a2},T_{f1}),\mathrm{MAX}\ (T_{g6}),$

$\mathrm{MAX}\ (T_{g7} + T_{g8},T_{d5},T_{e5},T_{f3}),\mathrm{MAX}\ (T_{g9},T_{h6}),\mathrm{MAX}\ (T_{e6},T_{f6},T_{g10},T_{b10}))$

$\mathrm{arriveRM16} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{i2},T_{c3} + T_{c4},T_{a2},T_{f1}),\mathrm{MAX}\ (T_{g6}),$

$\mathrm{MAX}\ (T_{g7} + T_{g8},T_{d5},T_{e5},T_{f3}),\mathrm{MAX}\ (T_{g9},T_{h6}),\mathrm{MAX}\ (T_{e6},T_{f6},T_{g10},T_{b10}))$

$\mathrm{arriveRM18} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{i2},T_{c3} + T_{c4},T_{a2},T_{f1}),\mathrm{MAX}\ (T_{g6}),$

$\mathrm{MAX}\ (T_{g7} + T_{g8},T_{d5},T_{e5},T_{f3}),\mathrm{MAX}\ (T_{g9},T_{h6}),\mathrm{MAX}\ (T_{e6},T_{f6},T_{g10},T_{b10}))$

$\mathrm{arriveRM2} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{b1},T_{c2}),\mathrm{MAX}\ (T_{g2},T_{h1},T_{b2},T_{g3}),$

$\mathrm{MAX}\ (T_{g6}),\mathrm{MAX}\ (T_{a5},T_{b6},T_{d4},T_{i5}),\mathrm{MAX}\ (T_{h5},T_{f2}),\mathrm{MAX}\ (T_{d7},T_{f4} + T_{f5},T_{b7},T_{d8}))$

$\mathrm{arriveRM5} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{b1},T_{c2}),\mathrm{MAX}\ (T_{g2},T_{h1},T_{b2},T_{g3}),$

$\mathrm{MAX}\ (T_{g6}),\mathrm{MAX}\ (T_{a5},T_{b6},T_{d4},T_{i5}),\mathrm{MAX}\ (T_{h5},T_{f2}),\mathrm{MAX}\ (T_{d7},T_{f4} + T_{f5},T_{b7},T_{d8}))$

$\mathrm{arriveRM11} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{b1},T_{c2}),\mathrm{MAX}\ (T_{g2},T_{h1},T_{b2},T_{g3}),$

$\mathrm{MAX}\ (T_{g6}),\mathrm{MAX}\ (T_{a5},T_{b6},T_{d4},T_{i5}),\mathrm{MAX}\ (T_{h5},T_{f2}),\mathrm{MAX}\ (T_{d7},T_{f4} + T_{f5},T_{b7},T_{d8}))$

$\mathrm{arriveRM12} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{b1},T_{c2}),\mathrm{MAX}\ (T_{g2},T_{h1},T_{b2},T_{g3}),$

$\mathrm{MAX}\ (T_{g6}),\mathrm{MAX}\ (T_{a5},T_{b6},T_{d4},T_{i5}),\mathrm{MAX}\ (T_{h5},T_{f2}),\mathrm{MAX}\ (T_{d7},T_{f4} + T_{f5},T_{b7},T_{d8}))$

$\mathrm{arriveRM15} = \mathrm{MAX}\ (\mathrm{MAX}\ (T_{b1},T_{c2}),\mathrm{MAX}\ (T_{g2},T_{h1},T_{b2},T_{g3}),$

$\mathrm{MAX}\ (T_{g6}),\mathrm{MAX}\ (T_{a5},T_{b6},T_{d4},T_{i5}),\mathrm{MAX}\ (T_{h5},T_{f2}),\mathrm{MAX}\ (T_{d7},T_{f4} + T_{f5},T_{b7},T_{d8}))$

Figure 3.14: Arrival time by maximum parts processing time in a product.

$$
\begin{aligned}
&\text{arriveRM1} = \text{MAX}\,(T_{a1},T_{c1})\\
&\text{arriveRM2} = \text{MAX}\,(T_{b1},T_{c2})\\
&\text{arriveRM3} = \text{MAX}\,(T_{d1},T_{g1},T_{i1})\\
&\text{arriveRM4} = \text{MAX}\,(T_{i2},T_{c3} + T_{c4},T_{a2},T_{f1})\\
&\text{arriveRM5} = \text{MAX}\,(T_{g2},T_{h1},T_{b2},T_{g3})\\
&\text{arriveRM6} = \text{MAX}\,(T_{e1},T_{g4},T_{c5},T_{b3},T_{h2})\\
&\text{arriveRM7} = \text{MAX}\,(T_{a3},T_{d2} + T_{d3},T_{e2},T_{g5},T_{h3})\\
&\text{arriveRM8} = \text{MAX}\,(T_{g6})\\
&\text{arriveRM9} = \text{MAX}\,(T_{h4},T_{i3},T_{b4},T_{e3})\\
&\text{arriveRM10} = \text{MAX}\,(T_{e4},T_{b5},T_{a4},T_{c6},T_{i4})\\
&\text{arriveRM11} = \text{MAX}\,(T_{a5},T_{b6},T_{d4},T_{i5})\\
&\text{arriveRM12} = \text{MAX}\,(T_{h5},T_{f2})\\
&\text{arriveRM13} = \text{MAX}\,(T_{g7} + T_{g8},T_{d5},T_{e5},T_{f3})\\
&\text{arriveRM14} = \text{MAX}\,(T_{a6},T_{c7},T_{d6})\\
&\text{arriveRM15} = \text{MAX}\,(T_{d7},T_{f4} + T_{f5},T_{b7},T_{d8})\\
&\text{arriveRM16} = \text{MAX}\,(T_{g9},T_{h6})\\
&\text{arriveRM17} = \text{MAX}\,(T_{c8},T_{b8},T_{a7},T_{b9})\\
&\text{arriveRM18} = \text{MAX}\,(T_{e6},T_{f6},T_{g10},T_{b10})\\
&\text{arriveRM19} = \text{MAX}\,(T_{f7},T_{d9},T_{b11},T_{g11},T_{h7},T_{i6})\\
&\text{arriveRM20} = \text{MAX}\,(T_{c9},T_{d10},T_{g12})
\end{aligned}
$$

Figure 3.15: Arrival time by single part processing time.

(B) WIP control

Work in progress (WIP) consists of unfinished products in the production process. Production management aims to minimize work in progress as WIP requires storage space, represents capital tied-up and stagnant production flow. While Just-in-time (JIT) production is ideal for WIP control, it is extremely difficult to achieve in a dynamic low volume manufacturing environment. In WITNESS simulation model, there is no direct mechanism to achieve the WIP control. The WIP level depends on the processing scheduling and input output rules. However, the WIP level can be monitored by 'NWIP' function in WITNESS.

The NWIP( ) function in WITNESS returns an integer value containing the number of parts of the specified type (specifies in the brackets) that are still progressing through the model.

In order to control the WIP level in the model, an 'if statement' control logic is used on the first processing machine of each parts when the parts are just pulled into the processing system. The statement is illustrated in Figure 3.16. The *wip* is a user controlled variable which is set before the simulation starts. For example, if the user sets *wip* to 6, the processing machine will only pull the parts into the process if the NWIP function returns a number less than 6. In this way, the level of WIP can be controlled by the simulation user.

```
IF NWIP (RM1) < wip
PULL from RM1 out of BufferA01(1)
ELSE
Wait
ENDIF
```

Figure 3.16: WIP level of Raw Material 1.

(C) Operator rules

All machines in the model require an operator to perform the processing tasks. Operator rules specify the type and quantity of operator needed to complete the task. In a machine general detail dialogue (Cf. Figure 3.4), operator rules can be edited in the *Duration* section as shown in Figure 3.17.

The basic operator rule is set up simply by entering the name of the operator required by a machine. For instance, Machine A01 needs an OperatorA for the processing function. All seven machine elements (Machine A01 to A07) are specified with OperatorA and there is only one Operator A in the system. Hence, there is only one machine working amongst the seven machine elements in the model at any one time. This makes it an ideal logical representation of one physical machine. With seven machine elements in the model there are seven control platforms for different parts which lead to easier and clean computer modelling.

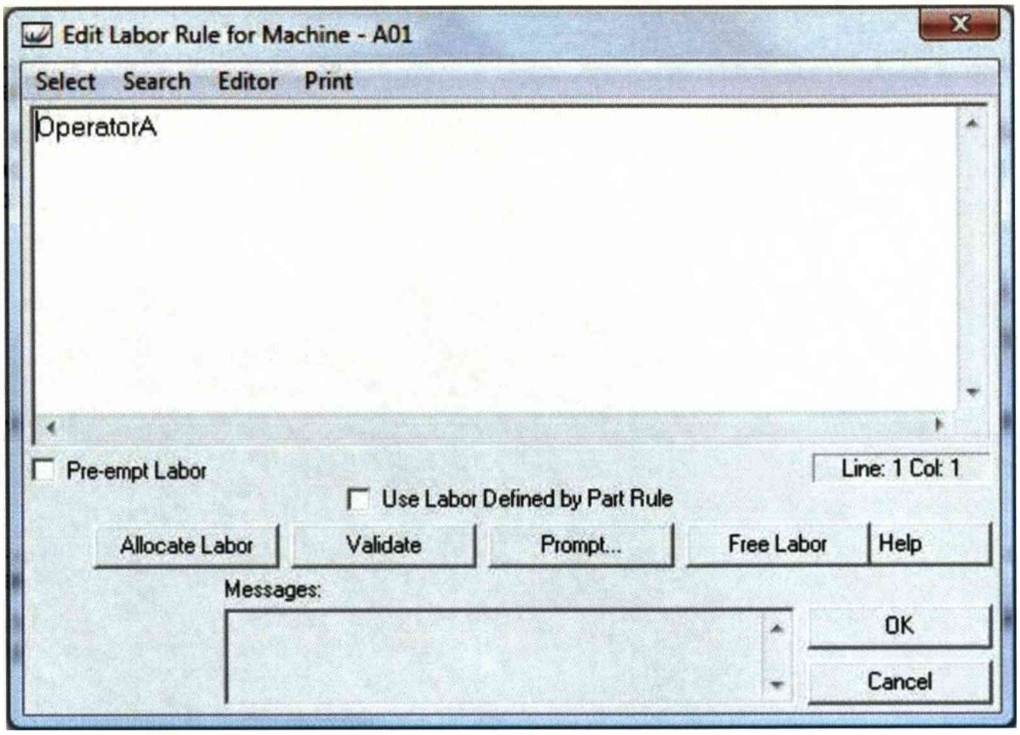


Figure 3.17: Labour rules edit window for machine elements.

More details can be specified as pre-empt operator rules by ticking 'Pre-empt labour' in the above figure. Then more rules can be given for switching jobs between machine elements. This concept is not used in this model, as it assumes that all set-up time is zero, hence it is not necessary to analyse pre-empt unfinished parts. A processing task has to finish before the machine element release the operator to another machine element in this model.

(D)Kanban

Kanban is a concept created for Just-In-Time (JIT) scheduling rule. It is a means by which JIT is achieved (Ohno, 1988). Kanban is a signalling system to trigger actions. Kanban historically uses cards to signal the need for an item which is an effective tool to support the running of a production system. The principles of Kanban are transferred into simulation language and help the development of the logic of this simulation model.

An integer variable named '*kanban*' is set up in the model. The priority of products for P1 to P5 is calculated and ranked 1 to 5, where 1 has the highest priority. Then *kanban* is used to flag the current priority to guide the processing of machines according to the priorities.

```
priority (1) = PriorityP1
priority (2) = PriorityP2
priority (3) = PriorityP3
priority (4) = PriorityP4
priority (5) = PriorityP5
SortVar (priority,1,0)
Relation (1) = IPOSRVAL (priority,PriorityP1,1,1)
Relation (2) = IPOSRVAL (priority,PriorityP2,1,1)
Relation (3) = IPOSRVAL (priority,PriorityP3,1,1)
Relation (4) = IPOSRVAL (priority,PriorityP4,1,1)
Relation (5) = IPOSRVAL (priority,PriorityP5,1,1)
Decision (1) = IPOSRVAL (Relation,1.0,1,1)
Decision (2) = IPOSRVAL (Relation,2.0,1,1)
Decision (3) = IPOSRVAL (Relation,3.0,1,1)
Decision (4) = IPOSRVAL (Relation,4.0,1,1)
Decision (5) = IPOSRVAL (Relation,5.0,1,1)
kanban = Decision (1)
```

Figure 3.18: Initialise kanban.

The list of simulation commands used to initialise the *kanban* variable is given in Figure 3.18. PriorityP1 to PriorityP5 are the values calculated for the five products. They can be determined with system data or simply assigned by customers and manufacturing managers under certain criterions such as the due date of shipping, the stock level of each product. The SortVar function sorts the array *priority* according to the value. The *Relation* array stores the sequence of the priorities and passes the name to *Decision* array. Initially, *kanban* is equal to *Decision(1)* which is the number of the product with highest priority. For instance, *kanban* is equal to 3 if product P3 has the highest priority.

```
MDemand (1) = Dp1 * week
MDemand (2) = Dp2 * week
MDemand (3) = Dp3 * week
MDemand (4) = Dp4 * week
MDemand (5) = Dp5 * week

IF Throughput (Decision (1)) < MDemand (Decision (1))
 kanban = Decision (1)
ELSEIF Throughput (Decision (2)) < MDemand (Decision (2))
 kanban = Decision (2)
ELSEIF Throughput (Decision (3)) < MDemand (Decision (3))
 kanban = Decision (3)
ELSEIF Throughput (Decision (4)) < MDemand (Decision (4))
 kanban = Decision (4)
ELSEIF Throughput (Decision (5)) < MDemand (Decision (5))
 kanban = Decision (5)
ENDIF
```

Figure 3.19: Changing kanban on the output machines.

Kanban will change after the completion of the product with the highest priority. The commands for controlling the *kanban* are shown in Figure 3.19. The command can be translated as: if the throughput of the highest priority product is equal to the market demand, *kanban* is set to the product with the next highest priority.

(E) Machine priority rules

The sequence of machine processing has to meet the Kanban system and act upon it. The priority setting in machine detail dialogue makes it possible.

Twenty real integers (PO1 to PO20) are created for the machine priority. Each part is allocated with one of these variables for the machine elements. For example, MachineA01 and MachineC01 will process RM1, so PO1 is employed in their priority field.

Priority variables are aligned with *kanban*. In other words, if *kanban* indicates product P1 has priority, then all machines which process parts for product P1 will have higher priority than other machines. With a higher priority, they can obtain operators and complete the processing before other machines.

3.6.7 Experimentation

When a simulation model is validated and verified, it resembles the behaviour of the real-life situation. A series of experiments are carried out to investigate a number of what-if scenarios. A typical experiment involves using a warm-up period or a set of starting conditions and a decision on a suitable run length.

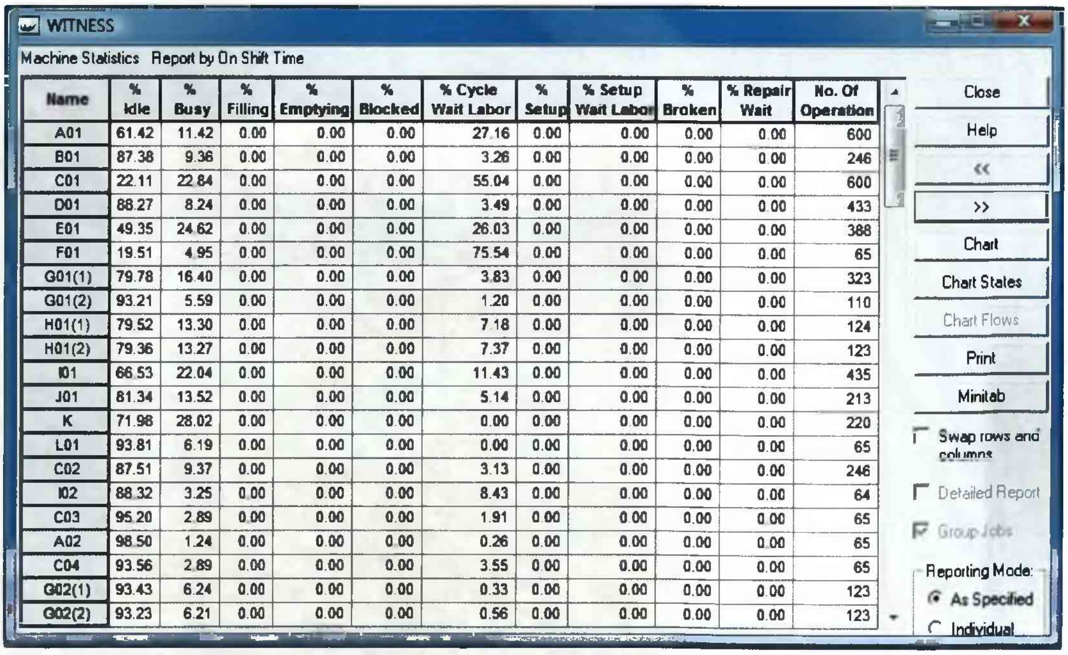
A warm-up period allows the model to reach a steady state before WITNESS collates any results. A warm up period of 30 minutes was chosen through experimentation with this value comparing the output of the model. After 30 minutes, all parts have entered the model and most of machines are in use. Hence the system reaches a steady state. In addition, it was found that by varying this value between 5 minutes and 60 minutes, the maximum difference in output was 1 or 2 products. Generally speaking, the most important consideration would be to keep this value constant so that no variation to the output would result from changing this value. Therefore, a value of 30 minutes was used throughout all experiments.

WITNESS experimentation can be carried out in the WITNESS model directly or as an add-on software from WITNESS Simulation Suite. WITNESS Scenario Manager and WITNESS Optimizer are both excellent tools for simulation experimentation. In this research, many model verification steps are completed in Scenario Manager. It is a powerful tool to set up and run simulation experiments and to obtain reports and charts across different scenarios. More details on optimisation experiments are given in Section 4.5.

3.6.8 Output results

Once the model has run, WITNESS generates reports such as statistics reports, detailed statistics report, user reports, summary reports and explode reports. In the middle of a running model, current status lists can be generated showing the present conditions in the model. Reports allow a user to examine the performance of elements in the model and provide a user with relevant information about their interaction, details and status. Nearly all important information can be found in the reports. It is therefore possible to gather the experiment reports for analysis to find out whether the objectives of the experiment have been met.

Among all the reports, statistics reports are most used and contain the greatest amount of useful information. It provides a statistical overview of the performance of all elements in the model. For example, Figure 3.20 shows a machine statistic report which gives percentage time on idle, busy, filling, emptying, blocked and so on. It monitors and records all activities for all elements.



WITNESS

Machine Statistics Report by On Shift Time

Name	% Idle	% Busy	% Filling	% Emptying	% Blocked	% Cycle Wait Labor	% Setup	% Setup Wait Labor	% Broken	% Repair Wait	No. Of Operation
A01	61.42	11.42	0.00	0.00	0.00	27.16	0.00	0.00	0.00	0.00	600
B01	87.38	9.36	0.00	0.00	0.00	3.26	0.00	0.00	0.00	0.00	246
C01	22.11	22.84	0.00	0.00	0.00	55.04	0.00	0.00	0.00	0.00	600
D01	88.27	8.24	0.00	0.00	0.00	3.49	0.00	0.00	0.00	0.00	433
E01	49.35	24.62	0.00	0.00	0.00	26.03	0.00	0.00	0.00	0.00	388
F01	19.51	4.95	0.00	0.00	0.00	75.54	0.00	0.00	0.00	0.00	65
G01(1)	79.78	16.40	0.00	0.00	0.00	3.83	0.00	0.00	0.00	0.00	323
G01(2)	93.21	5.59	0.00	0.00	0.00	1.20	0.00	0.00	0.00	0.00	110
H01(1)	79.52	13.30	0.00	0.00	0.00	7.18	0.00	0.00	0.00	0.00	124
H01(2)	79.36	13.27	0.00	0.00	0.00	7.37	0.00	0.00	0.00	0.00	123
I01	66.53	22.04	0.00	0.00	0.00	11.43	0.00	0.00	0.00	0.00	435
J01	81.34	13.52	0.00	0.00	0.00	5.14	0.00	0.00	0.00	0.00	213
K	71.98	28.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	220
L01	93.81	6.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	65
C02	87.51	9.37	0.00	0.00	0.00	3.13	0.00	0.00	0.00	0.00	246
I02	88.32	3.25	0.00	0.00	0.00	8.43	0.00	0.00	0.00	0.00	64
C03	95.20	2.89	0.00	0.00	0.00	1.91	0.00	0.00	0.00	0.00	65
A02	98.50	1.24	0.00	0.00	0.00	0.26	0.00	0.00	0.00	0.00	65
C04	93.56	2.89	0.00	0.00	0.00	3.55	0.00	0.00	0.00	0.00	65
G02(1)	93.43	6.24	0.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	123
G02(2)	93.23	6.21	0.00	0.00	0.00	0.56	0.00	0.00	0.00	0.00	123

Figure 3.20: Machine statistics report.

3.6.9 Conclusions

Simulation models benefit from continuous development and verification. A model can be developed with more refinements to become closer to a real-life situation. The end point of any simulation model development is to be able to predict the actions of a real-world mechanism. The level to which the model is developed is a function of the fidelity required in order to make the simulation model fit for purpose. This twenty part twelve machine 20P/12M model certainly has potential for further improvement. However, one of the principles of building a simulation model is that the complexity should coincide with the purpose of the simulation project. The model can give most the results required based on a given set of assumptions.

Chapter 4
Manufacturing System Reconfiguration

4.1 Introduction

This chapter presents a series of manufacturing system reconfiguration improvements. The object-oriented simulation model demonstrated in Chapter Three is used as the studied manufacturing system. The approach of detecting system reconfiguration requirements, locating the bottleneck work station or machine, deciding the level of reconfiguration as well as proposed methods to complete the reconfiguration are explained and illustrated in this chapter. Section 4.2 introduces the importance and contribution of the reconfigurable manufacturing system. Section 4.3 explains how to use Product Life Cycle to generate market demand pattern for inputs to the simulation model. Section 4.4 uses the Theory of Constraints to identify the bottlenecks for each configuration. Section 4.5 explains reconfiguration requirements brought about due to the changing market demand. Machine reconfiguration results are listed in Section 4.6 and further discussed in Section 4.7. Finally, Section 4.8 shows the impact of different scheduling rules.

4.2 Concept and importance of manufacturing reconfiguration

Competition in manufacturing industry will put greater endeavour on the ability to respond to the ever-changingmarket demand rapidly and flexibly. Due to significantly shortened product life cycles, manufacturers have found that they can no longer capture market share and gain high profit margin by maintain their status quo. It is essential for a manufacturer to be adaptive to changes in market in which they operate through almost constant reconfiguration. Success in manufacturing requires the adoption of methods in customer acquisition and order fulfilment processes that can manage anticipated change with precision while providing a fast and flexible response to unanticipated changes (Fulkerson, 1997). Customer demands should be incorporated in system design strategies for the reconfiguration of production facilities in order to sustain competitiveness and agility in an unpredictable business environment.

4.2.1 Scalable RMS

A reconfigurable manufacturing system (RMS) that is designed specifically to adapt to changes in production capacity, through system reconfiguration is called a scalable-RMS. The set of system configurations that a scalable-RMS assumes as it changes over time is called its configurations path (Patrick & Carlo, 2007).

The concept of scalability in manufacturing system is dated to the early 1980s (Browne 1984, Sethi, 1990). Expansion flexibility was the term used in the literature of flexible machining systems (FMS). Literally speaking, expansion flexibility is defined as the capability to expand or contract the production capacity of an FMS using a modular structure. At that time, studies on expansion flexibility was limited to general concept and characteristics of such systems rather than design details or guidelines. Sethi summarized recommendations for design of manufacturing systems with expansion flexibility (Sethi, 1990). The recommendations with implications on machine design include the following:

- Building small production units and expand by duplicating these small units.
- Provide an infrastructure to facilitate system growth.
- Design the system so it can be expanded without requiring significant new designs.

The above recommendations are the general characteristics found in the literature at that time. They neither establish nor intend to represent a comprehensive systematic design approach applicable to the Scalable RMS technology. Therefore, there are hardly any examples of such systems or system design methodologies.

Nowadays, there are many computerised numerically controlled (CNC) machines and reconfigurable machine tools (RMT) related RMS system design case studies. The majority of these studies propose a modular based reconfigurable system or reconfigurable machine. In particular, the research work carried out at the University of Michigan addresses system-level design issues in scalable RMSs. Koren et al. (Koren, et al. 1998), Spicer et al. (Spicer, et al. 2005) concluded that parallel configurations with crossover yield significant benefits in throughput performance and scalability when identical machines are used throughout the system. Parallel configurations allow portions of the system to continue operating even when new equipment is being added. Furthermore, duplication of existing processes and equipment in parallel minimizes additional engineering design effort.

Scalable machines are important concepts and design factor for realising scalable systems. The concept of a reconfigurable machine tool is introduced as a modular machine with a modular control system that gives manufacturers the ability to modify machine structure to satisfy specific manufacturing needs (Landers, 2001). Spicer et al. designed a multi-spindle machine tool specifically for scalability (Spicer, et al. 2002). This design concept provides the option of adding and removing multiple spindles on an as-needed basis. The benefits of this concept include: reduced capital investment, reduced reconfiguration time, and reduced consumption of space.

As described in Section 2.2, scalability is listed as one of six key characteristics of a reconfigurable manufacturing system (Koren, et al. 1999). The definition given by Koren is 'the ability to adjust the production capacity of a system through system reconfiguration with minimal cost, in minimal time, over a large capacity range, at given capacity increments'.

4.2.2 Capacity scalability

Capacity scalability is basically the ability to adapt to changing demand. Many manufacturers have difficulty purchasing capacity on an as-needed basis because changes in market demand and customers requirements out-pace system design and configuration. A scalable reconfigurable manufacturing system is the solution for adapting efficiently to changes in capacity requirements through system reconfiguration. In a scalable RMS, rapid modifications in system and machine structures through the addition and removal of productive equipment is the way to achieve the as-needed capacity increases to respond to market and demand fluctuations in minimal time.

A typical capacity scalability problem addresses when, where and by how much should the capacity of the manufacturing system be scaled. The revolution of scalable RMS is shown in two major principles (Deif & Elmaraghy, 2007). One is that capacity scalability is not limited to increases. Capacity is scalable for both volume increase and reduction. Secondly, capacity scalability is not only over the system level but also over the machine level by adding or removing machine modules on open control structures. In this investigation, the term capacity scalability is used to describe the capacity expansion during system reconfigurations.

An approach for modelling capacity scalability is proposed. Unlike most of the other RMS researches which focus on the design of new reconfigurable machine tools, this research uses simulation software to model the whole system rescaling process. Based on the simulation model, a decision making system (DMS) that utilises the simulated annealing optimisation technique is developed. The DMS can be used to aid systems designers in deciding when, where and how much to reconfigure the system in order to meet the market demands in a cost-effective way.

Figure 4.1 illustrates the process of developing a decision making system for manufacturing system. Notice the difference here is that no RMT is required in the system. This makes it possible to implement in almost any manufacturing enterprises.

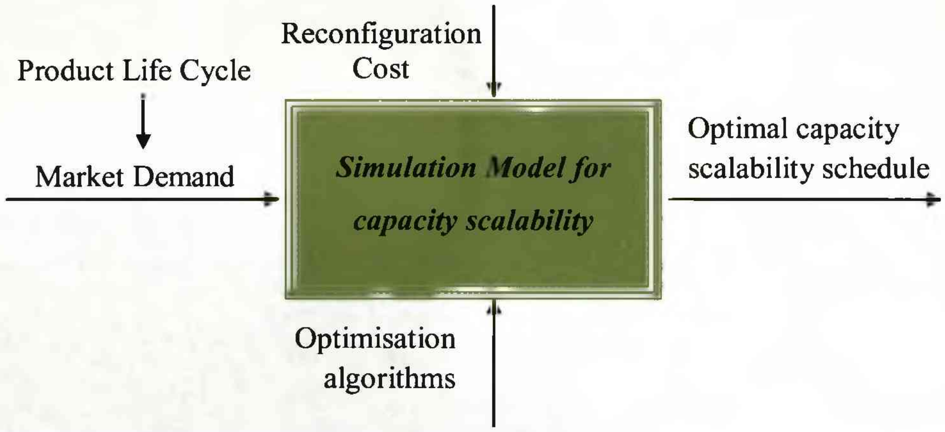


Figure 4.1: Reconfigurable manufacturing DMS development process.

4.3 Using product life cycle to predict demand trend

One common nomenclature which considers market demand over time is the Product Life Cycle (PLC). PLC is used as the measure of units sold or demanded over the life time of a product. PLC is based on the biological life cycle. In theory, the genetic life cycle of a product is the same. After a period of development it is introduced or launched into the market; it gains more and more customers as it grows; eventually the market stabilises and the product becomes mature; then after a period of time the product is overtaken by development and the introduction of superior products, it goes into decline and is eventually withdrawn. This definition points out the PLC deals with both demand and time dimensions for a product.

Since the first introduction of the PLC concept by Levitt, the characteristics for each stage of the cycle from introduction, growth, and maturity and decline have been studied in great detail (Levitt, 1965). It was noted that each stage required its own distinctive manufacturing strategy and competitive stance as a result of different customer expectations. This also infers that reconfiguration of manufacturing is essential to deal with the change of demand over time. Therefore, PLC is adapted to model a general, time-varying, deterministic demand pattern. The demand pattern follows the product life cycle shape which is used as the market demand input of the RMS simulation model.

4.3.1 Discrete demand pattern

This research studies the demand through a complete product life cycle but its approach is not restricted to the PLC context. The demand curve has been broken up into four segments and each segment is approximated into a discrete quantity.

The assumption that the demand pattern is uniform over a time period is likely to be a dubious one in many practical contexts. However, it suits the circumstances of this discrete event simulation study. The demand input of the simulation model is constant within each simulation period and varies between them to follow the product life cycle trend. In this investigation, the time period was chosen on a quarterly basis. The whole simulation period covers five quarters which is long enough to reflect the whole product life cycle under the assumption that the product life is approximately one year. Discretisation was applied to each time period of the product life cycle. Figure 4.2 shows the expected trend of a product life cycle and the smoothed demand level in each stage.

The demand pattern apparently simplified the practical situations for most products. It can only be justified by the fact that a full analysis based on a time-varying stochastic demand process would be too mathematically and computationally intensive. In addition, there are five products in the simulation model. They are in different development stage within the life cycle. Adding together five random demand functions for each product, it would extend simulation time and reduce the confidence of simulation results. Furthermore, for practical applications, the product life cycle time scale and demand quantities can be adjusted at a later stage. First hand data can be obtained from sales offices; more accurate demand input could be used to feed the simulation model to generate reconfiguration decision in a relatively shorter period, for instance, on a monthly or weekly timescale.

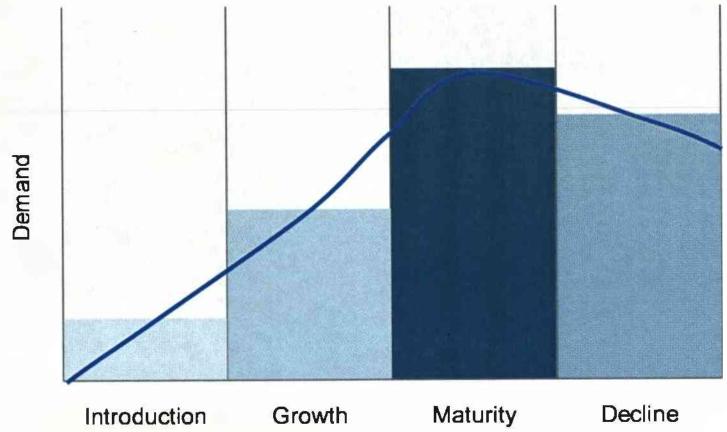
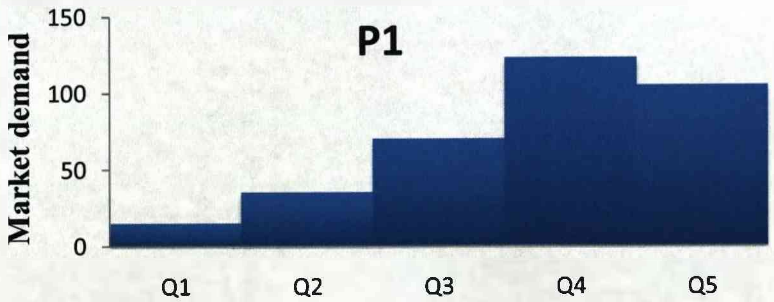


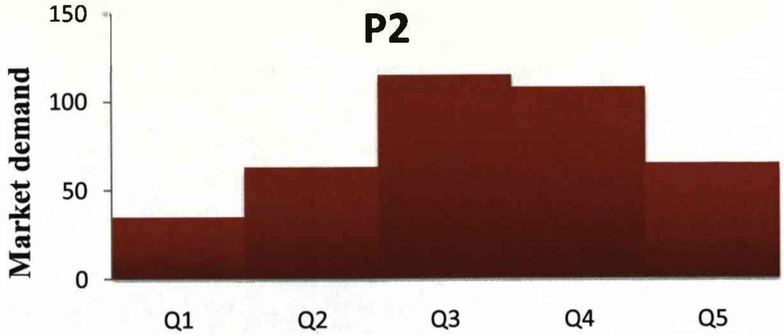
Figure 4.2: Product life cycle showing variable demand.

4.3.2 Demand patterns for the five products

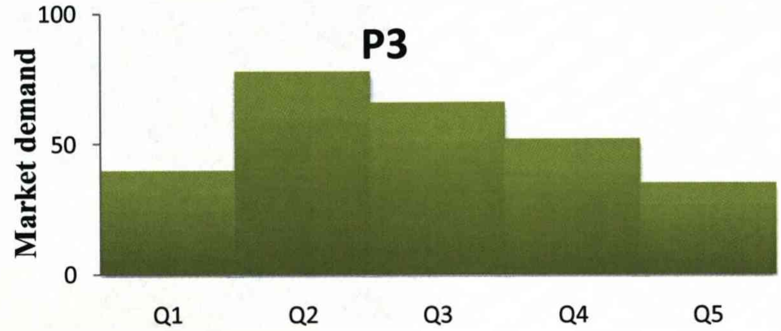
For a multi-stage multi-product line, the overall production situation is complex as the demand for each product is time-variant. There are five products in the simulation model of this research. The life cycle for each product was chosen at random from the introduction phase to decline phase over this period and the level of demand for each product is different. The mix of the differing life cycle profiles are intended to emulate a heterogeneous real world situation. The profiles can be changed simply by adjusting the input file for each simulation run.



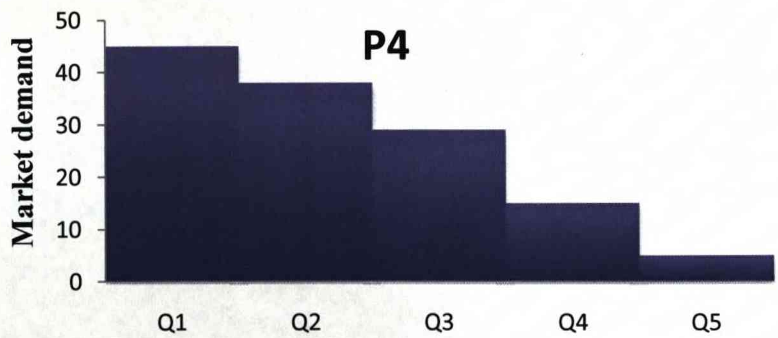
(a)



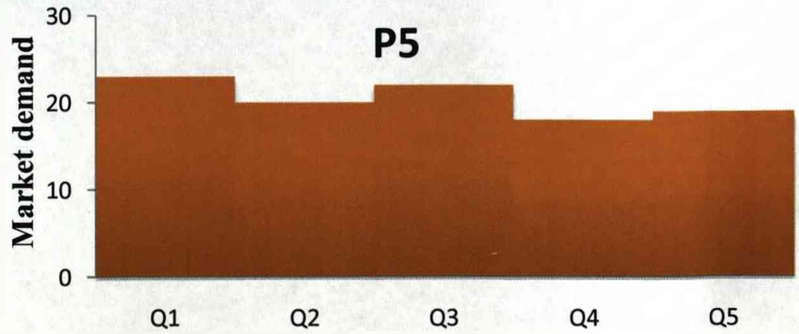
(b)



(c)



(d)



(e)

Figure 4.3: Market demand pattern for five products.

Figure 4.3 shows the weekly market demand pattern for product P1 to P5. All five products are at different stages of product life cycle. In each quarter, it is assumed that weekly demand for each product is constant. In other words, the simulation model reads the demand change on a quarterly basis. Product P1 shows a typical product life cycle from introduction to decline. Product P2 is slightly further into the product life cycle at quarter one, therefore, there is a quite significant decline of P2 demand in quarter five. P3 starts from a growth period then becomes mature in quarter two and declines from quarter three. P4 is already a mature product from quarter 1. Decline in demand happens gradually every quarter. From the demand pattern of product P5, it is hard to tell which cycle period it is in as the demand fluctuation. It could be due to unpredictable change of the market or the result of sales promotion.

Table 4.1: Market demand of the five products for each demand period.

Product	Q1	Q2	Q3	Q4	Q5
P1	195	455	910	1599	1365
P2	455	819	1495	1404	845
P3	520	1014	858	676	455
P4	585	494	377	195	65
P5	299	260	286	234	247

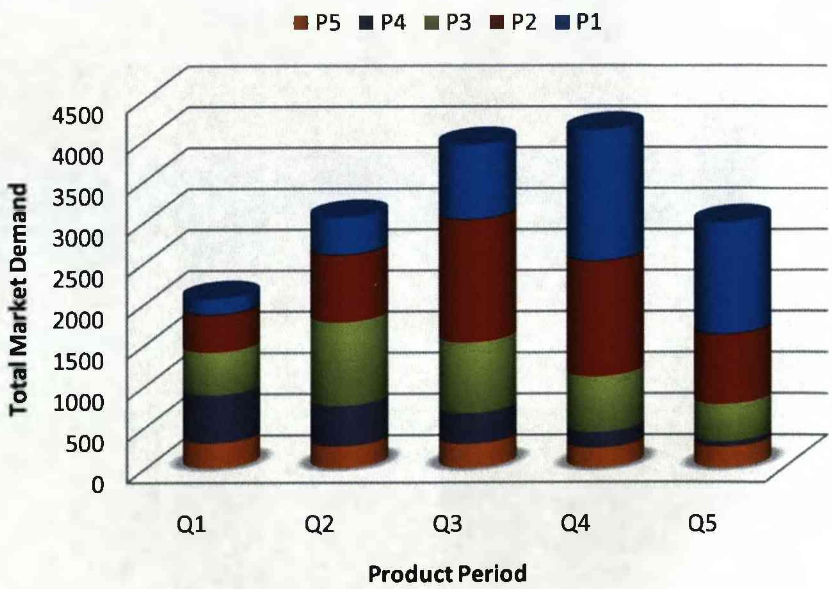


Figure 4.4 : Aggregated product demand pattern.

Table 4.1 gives the demand quantity of the five products from Q1 to Q5. The aggregate demand for the entire period is given in Figure 4.4. The data is generated according to the scale of the manufacturing model and the simplified product life cycle model. Demand inputs shown above are the result of an iterative process which is relatively realistic for this research.

4.4 Theory of Constraints (TOC) for bottleneck identification

The Theory of Constraints (TOC) as a philosophy for business, management and manufacturing has been extensively developed and provide a thinking process for managing a factory (Luebbe and Finch, 1992). It is an extremely useful practical tool to indentify the weakest link, commonly known as the bottleneck.

Goldratt stated that the system output rate was limited by the slowest rate of any machine (Goldratt, 1990). This is because in TOC there are only two types of machines: bottleneck machines or non-bottleneck machines. Bottleneck machines are the system constraints capping the ability of producing more products. The five well-known steps to elevate system constraints have been discussed in Section 2.3 and can be also interpreted in the following flow chart in Figure 4.5.

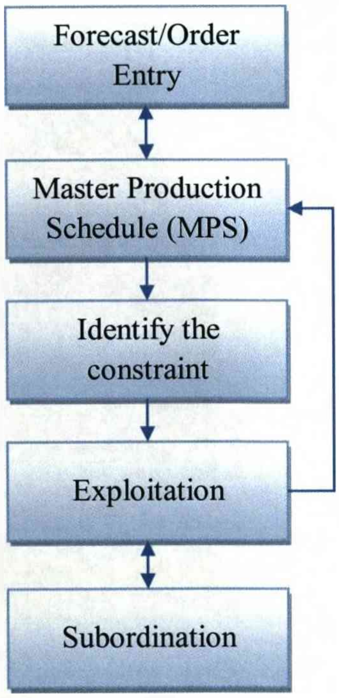


Figure 4.5: Steps for implementing TOC (Stein, 2003).

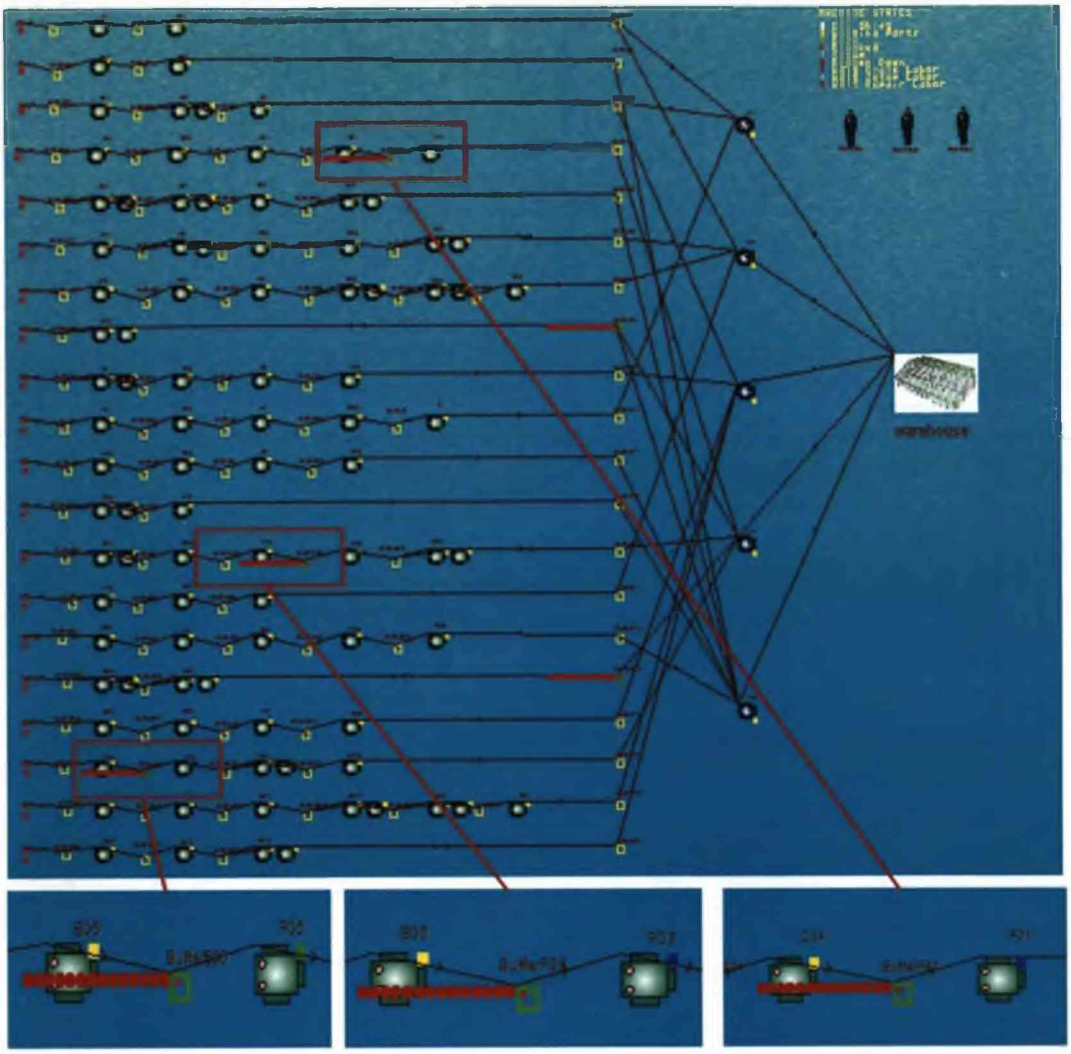
The first and the most important step of TOC implementation is to identify the system constraint. Without identifying the bottleneck machine, the following steps could not be implemented to improve the overall system performance. Theoretically, the identification is based on the relationship between machine capacity and capacity demand. In other words, if the resource operates at 100% utilisation, and there is still more work required on that resource, the machine is regarded as the bottleneck machine or system constraint. However, when there is more than one bottleneck machine in the manufacturing system, the identification could become complex. There are three simulation based methods to identify bottleneck machine candidates.

Firstly, bottleneck machine can be identified by observations of the processing queue. This method is also mentioned by Goldratt several times (Goldratt, 1990). In the simulation model the queue in front of bottleneck candidates can be easily detected. Figure 4.6 is a screen print of the simulation model at the end of quarter one. It can been seen from the figure that all machines have completed processing except Machine F. Red dots represent the number of parts queuing in front of Machine F. Therefore Machine F is the bottleneck machine and the system constraint in this configuration.

Secondly, machines that have the highest utilisation can be identified as bottleneck machines. WITNESS generated simulation statistics reports give machine utilisation percentage on busy, idle, blocked, set up etc. When the busy state percentage reaches 100%, this strongly suggests that the machine cannot complete the allocated operations.

The above two methods are very straightforward and easy to implement, however, in some complex job shop environments it is insufficient to identify the real bottleneck machine by these two methods and the conclusion may be misleading.

The last method is derived from the understanding of the definition of a bottleneck machine. Comparing the machines' available capacity with the required capacity associated with the market demand, if the available capacity of one machine is less than the machine time required to complete the operations to meet the market demand, the machine is defined as a bottleneck machine. It is the constraint of the system the bottleneck machine which prevents the system producing enough throughputs to meet the demand.



Red dots indicate the amount of queuing parts in buffers.

Figure 4.6: Identify bottleneck machines from the buffer queue.

The machine capacity requirement can be calculated by Equation 4.1:

$$C_R = \sum_{i=1}^{5} T_{P_i} D_{P_i} \qquad [4.1]$$

Where:

C_R = *the required machine capacity to meet market demand for all 5 products.*

T_{Pi} = *the cycle time the machine spends on processing parts for product P_i.*

D_{Pi} = *the weekly demand for product Pi.*

i = *1 to 5 as there are five products.*

Using Equation 4.1, the machine capacity requirement over five quarters can be calculated and displayed in Table 4.2.

Table 4.2: Weekly machine capacity requirement in minutes over five quarters.

Machine	Q1	Q2	Q3	Q4	Q5
A	849	1409	2221	2412	1692
B	1808	2758	3037	2879	2144
C	2133	3283	4421	4522	3134
D	1246	1947	2437	2282	1593
E	2315	3367	4479	4400	2910
F	2973	3062	2657	1827	1289
G	2200.5	2787	3071	2676	1796.5
H	2012.5	3024.5	3704	3175.5	2085
I	1854	3122	3384	3830	2992
J	570	1106	2080	2526	1830
K	800	1560	1320	1040	700
L	1066	910	809	531	398

The capacity requirement of machine G and H is for each machine although there are two G and two H in the system. For example, the total capacity required for machine G in Q1 is 4401 minutes, therefore each machine G has to provide 2200.5 minutes.

The available machine capacity is pre-determined according to the production schedule of the manufacturing system. In this study, the total working hour used to calculate the available capacity is typical in the UK manufacturing industry.

There are five working days per week, 8 hours shift per day and 60 minutes per hour. Therefore, the available cycle time per machine is $5 \times 8 \times 60 = 2400$ minutes as a theoretical maximum. For an actual shopfloor there must be shift changes, personal break time etc. that could be accommodated by the model easily. If the requirement for one machine is more than 2400 minutes, there is not enough time for this machine to complete the operation. Hence, such a machine is the bottleneck machine. The comparison is illustrated in Figure 4.7.

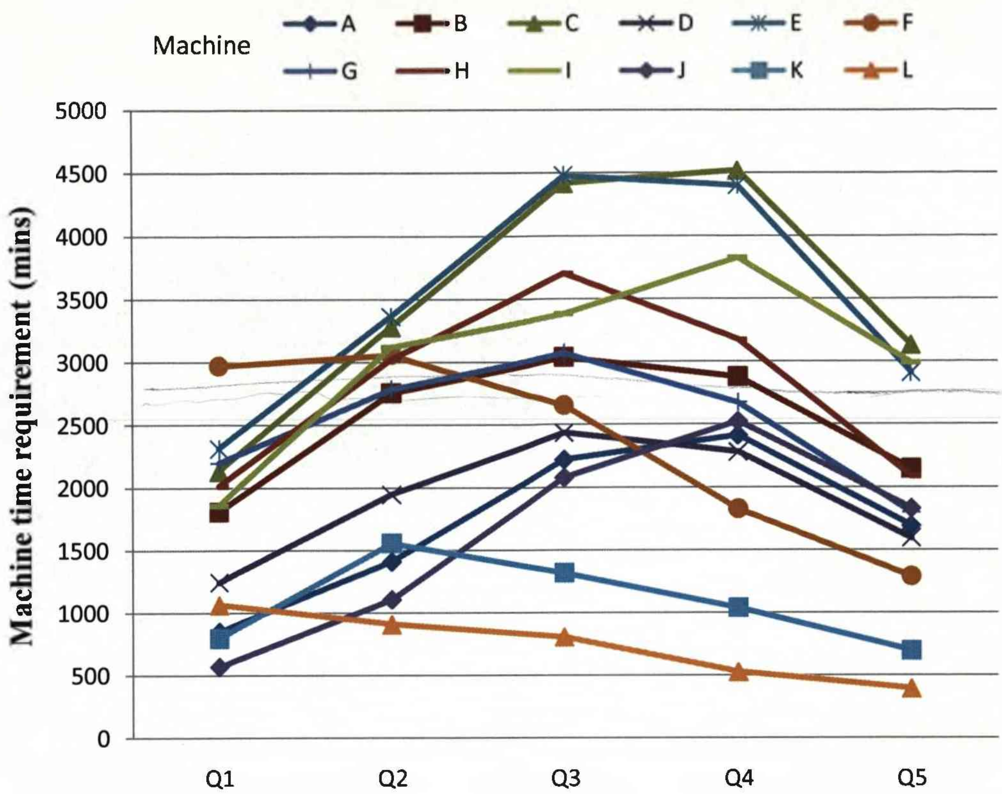


Figure 4.7: Machine capacity requirement over five quarters.

It is obvious that if over 2400 minutes is required from a machine, it is a bottleneck machine and there are quite a few bottleneck machines in Q2, Q3 and Q4. Figure 4.7 can also be converted to a percentage table by comparing the capacity requirement and the available capacity.

Table 4.3: Capacity requirement against available capacity.

	Q1	Q2	Q3	Q4	Q5
A	35.38%	58.71%	92.54%	100.50%	70.50%
B	75.33%	114.92%	126.54%	119.96%	89.33%
C	88.88%	136.79%	184.21%	188.42%	130.58%
D	51.92%	81.13%	101.54%	95.08%	66.38%
E	96.46%	140.29%	186.63%	183.33%	121.25%
F	123.88%	127.58%	110.71%	76.13%	53.71%
G	91.69%	116.13%	127.96%	111.50%	74.85%
H	83.85%	126.02%	154.33%	132.31%	86.88%
I	77.25%	130.08%	141.00%	159.58%	124.67%
J	23.75%	46.08%	86.67%	105.25%	76.25%
K	33.33%	65.00%	55.00%	43.33%	29.17%
L	44.42%	37.92%	33.71%	22.13%	16.58%

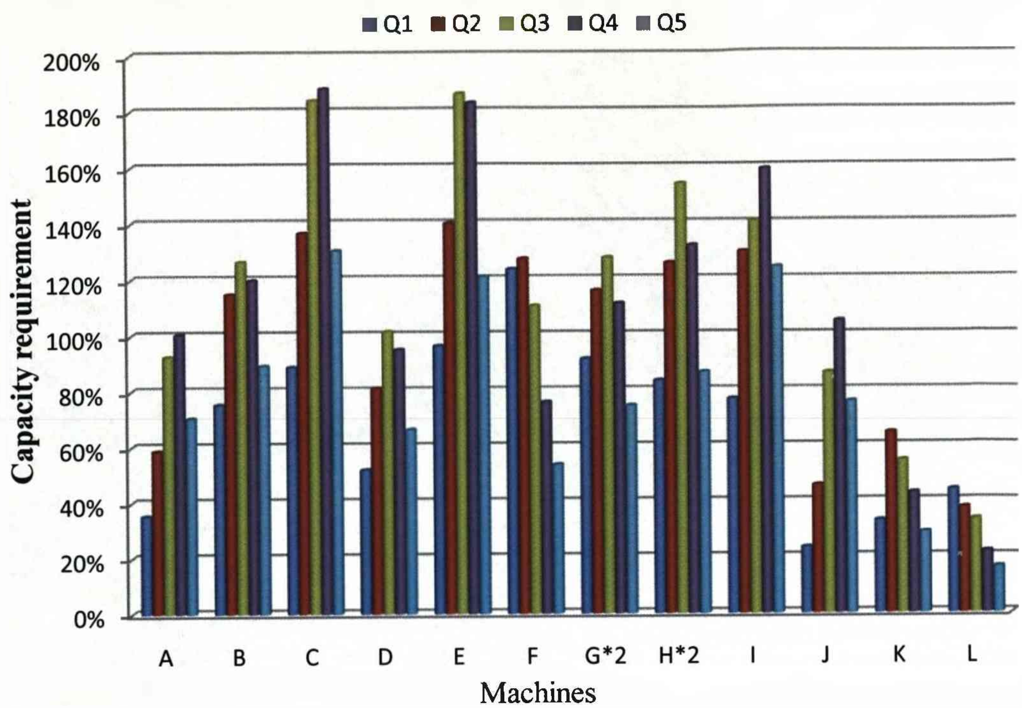


Figure 4.8: Capacity requirement.

Table 4.3 and Figure 4.8 give a more straightforward view of the capacity situation for each machine. For example, Machine A reaches 100% utilisation at Q4, and Machine B is a bottleneck machine at Q2, Q3 and Q4. As much as 180% of available capacity is required for Machines C and E in Q3 and Q4. The capacity requirements for Machines G and H are not very high because there are duplicate machines. This is very useful information for reconfiguration decision. The higher the machine capacity requirement, the more attention is required during reconfiguration.

4.5 Machine reconfiguration options

4.5.1 Options for manufacturing reconfiguration

An option is something available for selection, the power or freedom to choose or not to choose. Options become valuable when there is uncertainty in the manufacturing system and reconfiguration is required. As far as the reconfiguration of manufacturing system is concerned, options are referred to as those production options in accordance with reconfiguration technologies. Production planning decisions in general fall into three categories:

1) Short term, refers to day to day scheduling and sequencing.
2) Medium term, such as machine configurations and small scale change on system capacity.
3) Long term, involves total system layout, production line and production strategy.

Manufacturing decisions have to be made according to the different decision time scale so that proper reconfiguration technologies are chosen to maximize certain performance measures.

In this research, system configuration focuses on machine configuration which is a medium term decision. Short term decision with regard to scheduling rules is discussed in Section 4.8. The direct result of machine reconfiguration is the change of machine productivity. It is often referred to as machine capacity in the literature. In this context, machine capacity has two meanings. First, the available time for a machine to process jobs, i.e. 2400 minutes per week as mentioned earlier. Second, how fast does the machine work and how many parts can the machine process per week. Obviously, extending the machine shift time and adding overtime of the operator will give extra available production time, hence increase the machine capacity. In this research, only the second meaning, the machine production rate is taking into consideration. There are many ways to improve machine capacity, such as:

- The use of better tooling.
- Machine reconditioning.
- Replacement of the current machine by a new machine.
- The introduction of an additional machine, etc.

If the machines to be reconfigured are reconfigurable machines from the start, there are other ways to add extra capacity according to the design of the machines, such as adding or changing tools. In the simulation research, the cycle time the machine spends on processing a part is directly affected by the machine capacity improvements listed above. As a result of reconfiguration, the parameter machine cycle time (T_x) changes according to different levels of reconfiguration.

4.5.2 Simulation implementation for machine reconfiguration options

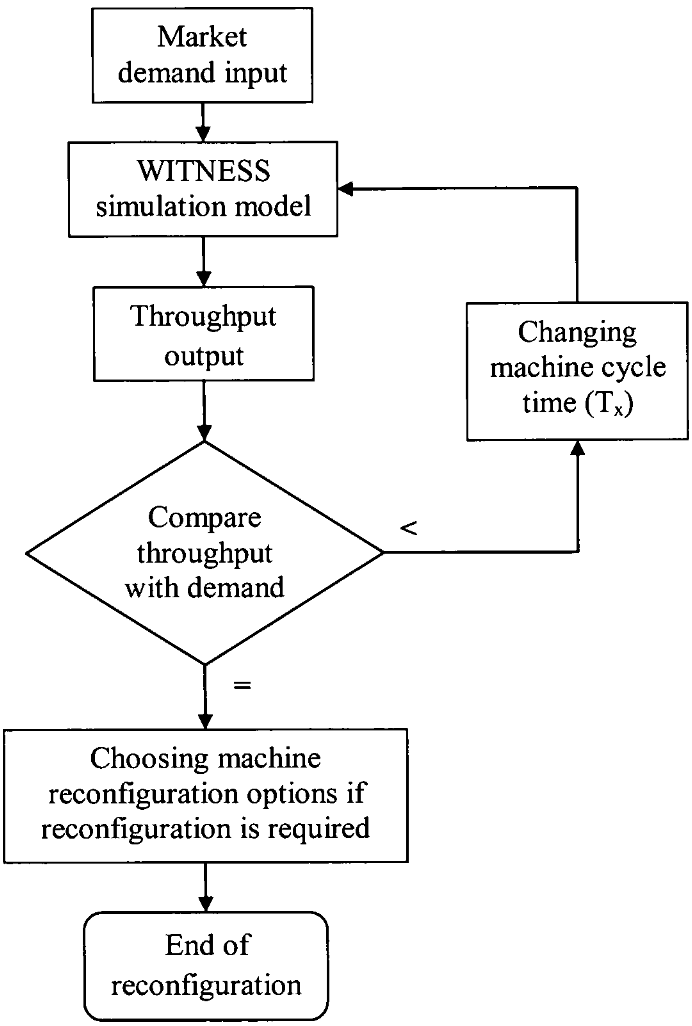


Figure 4.9: Sequence of reconfiguration options implementation.

The implementation sequence of machine reconfiguration is illustrated in the above flow chart. The market demand with product life cycle consideration is used as the input information for the WITNESS simulation model. The first throughput result is simulated with the current machine capacities (the original machine cycle times as given in Table 3.2). The comparison of market demand and system throughput will determine if reconfiguration is required. If the market demand is not met, a new calculated cycle time will be used to simulate the reconfiguration results until the throughput matches market demand. The new machine cycle time informs the level of reconfiguration, furthermore the reconfiguration option can be chosen to implement the reconfiguration decision.

4.5.3 WITNESS programming for machine reconfigurations

The following parameters are inserted into WITNESS to control the simulation of the machine configuration. Twelve parameters (PercentA to PercentL) represent the machine improvement profile inputs and are given as percentage improvements. PercentA is the improvement percentage of Machine A, PercentB is for Machine B and so on.

T_{xy} machine cycle time for material Y on machine X is the parameter originally applied in the machine X's detail dialogue of WITNESS model. In order to control the machine cycle time changes for reconfiguration, another twelve parameters, {a,b,c,d,e,f,g,h,ii,j,kk,l} are set up to calculate the relationship between machine cycle time and the percentage of improvement.

a = 1 / (1 + PercentA)

b = 1 / (1 + PercentB)

c = 1 / (1 + PercentC)

d = 1 / (1 + PercentD)

e = 1 / (1 + PercentE)

f= 1 / (1 + PercentF)

g = 1 / (1 + PercentG)

h = 1 / (1 + PercentH)

ii = 1 / (1 + PercentI)

j = 1 / (1 + PercentJ)

kk = 1 / (1 + PercentK)

l = 1 / (1 + PercentL)

*parameters i and k have been used by Witness as other default variables, so ii and kk were used instead.

For instance, if the capacity of machine A needs to improve by 20%, the system sets PercentA equal to 20%.

a = 1 / (1 + PercentA)=0.833

Hence, when the machine's capacity increases by 20%, the new machine cycle time equals to 0.833 times the original machine cycle time.

As shown above, it is quite straightforward to adjust the machine capacity by changing the machine cycle time. The twelve machine improvement percentage parameters (PercentA to PercentL) are stored in an input file named "Reconfiguration".

In reality, the machine configuration options should not be continuous numbers as there are not so many feasible choices to reconfigure the machines to x percent exactly. Therefore, a slightly higher improvement percentage result will be found so that the simulation throughput will meet the market demand. With the simulation results of the machine improvement requirements, practical reconfiguration decisions can be made along with the available reconfiguration options. A reconfiguration option which will reconfigure the system to the exact level of production capacity is unlikely to be found. A choice would have to be made. The solutions to these questions depend on the exact individual circumstances but the solutions must satisfy market demand as well as economic objectives. In practice, each case is unique and the actual properties of batch size, product mix, production sequences, space constraints and investment criteria will lead to different reconfiguration decisions.

4.6 Reconfiguration results

With reference to Table 4.3, Table 4.4 demonstrates the extra capacity required on bottleneck machines. As expected from the demand pattern, Q3 and Q4 have the biggest demand; hence, the system constraints reach the peak. There are up to eight bottleneck machines at Q3 and Q4. Only one bottleneck machine appears in Q1 as products P1, P2 and P3 are still in introduction or development phase. From the results of Table 4.4, the system production is constrained by bottleneck machines to a large extent. Therefore, this leads to the conclusion that system reconfiguration is totally necessary.

Table 4.4: Extra capacity required per quarter.

Machine	Q1	Q2	Q3	Q4	Q5
A				0.50%	
B		14.92%	26.54%	19.96%	
C		36.79%	84.21%	88.42%	30.58%
D			1.54%		
E		40.29%	86.63%	83.33%	21.25%
F	23.88%	27.58%	10.71%		
G		16.13%	27.96%	11.50%	
H		26.02%	54.33%	32.31%	
I		30.08%	41.00%	59.58%	24.67%
J				5.25%	
K					
L					

4.6.1 Simulation reconfiguration result and analysis

I. ***Quarter one results and analysis***

There is only one bottleneck machine in Q1, therefore it is relatively straightforward for reconfiguration. Machine F is the bottleneck and there is 23.88% extra capacity required for Machine F to complete the processing job in the available 2400 minutes per week.

Table 4.5: Quarter one machine utilisation before reconfiguration.

Machine	% Busy	% Idle
A	33.37	66.63
B	72.03	27.97
C	79.85	20.15
D	49.91	50.09
E	82.34	17.66
F	*100*	0
G	76.88	23.12
H	78.97	21.03
I	72.08	27.92
J	23.75	76.25
K	33.33	66.67
L	34.75	65.25

Table 4.6: Demand and throughput in Q1.

	P1	P2	P3	P4	P5
Throughput	195	455	520	*384*	299
Market Demand	195	455	520	585	299

Tables 4.5 and 4.6 are part of the statistics report for the current system configuration. Machine F reaches 100% utilisation and the system can only produce 384 of product P4 rather than the required 585 units. Simulation reconfiguration is carried out on the WITNESS Scenario Manage platform. PercentF is set as the variable of the experiment. It is set to run from 0% to 30% with an 1% interval. The results of thirty replications are given in Figure 4.10.

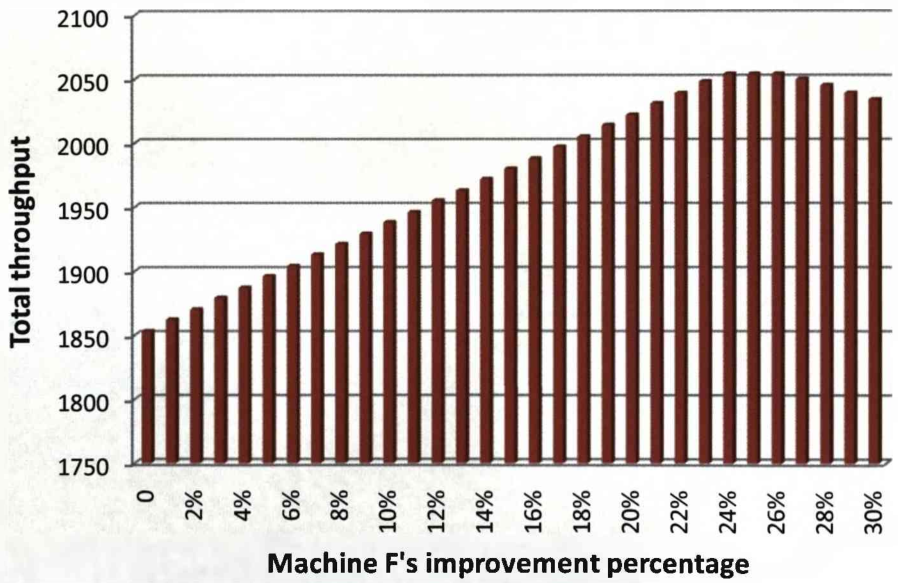


Figure 4.10: The total throughput change with the improvement of Machine F.

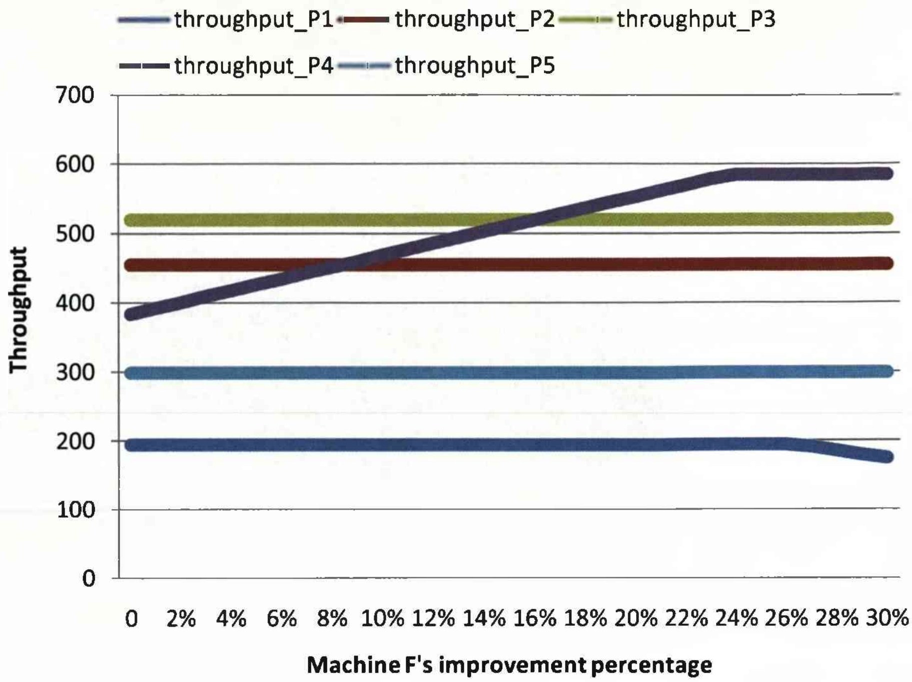


Figure 4.11: Throughput changes for five products.

The reconfiguration results shown in Figures 4.10 and 4.11 suggest an improvement of 23% to 26% on Machine F's capacity will remove the constraint on system production and produce enough products to meet the market demand. It is illustrated in Figure 4.11 the throughput of P4 is gradually going up with the Machine F's improvement percentage. These results again prove the TOC theory and demonstrate the relationship between constraint and production output.

The system production reaches the peak at 23% improvement of Machine F as the result of eliminating the bottleneck machine. No further improvement would be found since the system has met the market demand. However, production reduces with further improvement of Machine F which is out of expectation. Product P1 is reducing with extra improvement on Machine F. A small three machine simulation model is undertaken in order to explain the simulation results as given in Section 4.7.

II. *Quarter two results and analysis*

From the previous analysis there are a few bottleneck machines in Q2 due to a large increase in demand. Consequently, it is not easy to simulate the reconfiguration of machine improvement with many machines changes simultaneously. For the start of the reconfiguration process, a trial run is carried out where all machines reconfiguration levels are set according to the demand capacity as discussed under the Theory of Constraint.

Table 4.7: Required capacity for bottleneck machine in Q2.

Machine	Required extra capacity	Reconfiguration set to
B	14.92%	15%
C	36.79%	37%
E	40.29%	40%
F	27.58%	28%
G	16.13%	16%
H	26.02%	26%
I	30.08%	30%

Table 4.8: Machine utilisation before and after reconfiguration.

	Before		After	
Machine	% Busy	% Idle	%Busy	%Idle
A	52.78	47.22	55.66	44.34
B	95.05	4.95	96.01	3.99
C	100.00	0	96.51	3.49
D	75.87	24.13	80.11	19.89
E	100.00	0	100	0
F	100.00	0	99.58	0.42
G	98.34	1.66	100	0
H	98.52	1.48	95.5	4.5
I	100.00	0	94.2	5.8
J	10.81	89.19	25.35	74.65
K	33.87	66.13	49.87	50.13
L	26.74	73.26	37.92	62.08

Table 4.9: System throughput before and after reconfiguration.

		P1	P2	P3	P4	P5
Throughput	Before	199	567	698	363	260
	After	455	732	835	494	260
Market Demand		455	819	1014	494	260

Tables 4.8 and 4.9 show the changes of the machine utilisation and the system throughput before and after reconfiguration in Q2. There are seven bottleneck machines in this quarter according to the calculation based on TOC theory. Four of the bottleneck machines C, E, F and I reach 100% utilisation and three of them are reduced below 100% after reconfiguration. Machines B, G and H are bottlenecks as well, however, their utilisation do not reach 100% due to the process scheduling. It can be concluded that a bottleneck machine may not show 100% utilisation. Machines have 100% utilisation maybe not be a bottleneck either. Figure 4.12 illustrates the twelve machine utilisation changes before and after the reconfiguration. The effects of reconfiguration on bottleneck and non-bottleneck machines are dynamical. For instance, when the capacities of the bottleneck machines are increased after reconfiguration, their utilisations will decrease as expected. On the other hand, the pattern of utilisations for the non bottleneck machines shows the reverse trend as given in Figure 4.12. This is because non bottlenecks have to process more parts in order to feed the extra capacity now available in bottleneck machines. This means that more attention should be focused on those critical and near-critical machines.

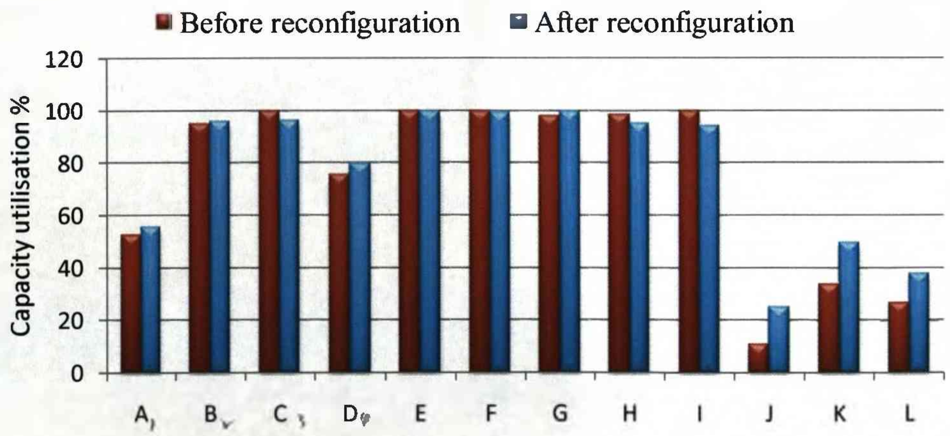


Figure 4.12: Machine utilisation before and after reconfiguration.

The system throughput has significant increases; both P1 and P4 meet market demand. However, from the above results, it can be observed that the system still cannot meet the market demand even after reconfiguration according to the TOC theory due to the complexity of scheduling.

III. Quarter three results and analysis

Tables 4.10 to 4.12 are the simulation inputs and results for Q3 reconfiguration. The scenario is very similar to Q2. There are eight bottleneck machines with even higher capacity requirement as the result of continuous increasing of market demand. Among them, two machines C and E require more than 80% extra capacity. It is suggested to add an extra machine to achieve a 100% improvement when the requirement for extra capacity is high as it is generally high cost and unfeasible to achieve a high capacity improvement by adding or changing machine tools. After a reconfiguration for eight bottleneck machines, all machine utilisations are below 100%, nevertheless there is still a gap between the system throughput and market demand due to machine scheduling problem.

Table 4.10: Required capacity for bottleneck machine in Q3.

Machine	Required extra capacity	Reconfiguration setting
B	26.54%	27%
C	84.21%	84%
D	1.54%	1.5%
E	86.63%	87%
F	10.71%	11%
G	27.96%	28%
H	54.33%	54%
I	41.00%	41%

Table 4.11: Machine utilisation before and after reconfiguration.

	Before		After	
Machine	**% Busy**	**% Idle**	**%Busy**	**%Idle**
A	71.10	28.90	91.52	8.48
B	90.08	9.92	96.20	3.80
C	100.00	0	97.71	2.29
D	87.34	12.66	98.21	1.79
E	100.00	0	99.70	0.30
F	100.00	0	99.64	0.36
G	100.00	0	96.35	3.65
H	96.69	3.31	93.54	6.46
I	100.00	0	72.45	27.55
J	15.19	84.81	40.33	59.67
K	48.13	51.88	25.95	74.05
L	29.38	70.62	30.06	69.94

Table 4.12: System throughput before and after reconfiguration.

		P1	P2	P3	P4	P5
Throughput	**Before**	220	581	667	348	286
	After	726	1236	771	377	286
Market Demand		910	1495	858	377	286

IV. Quarter four results and analysis

Q4's reconfiguration results show the similar pattern with the previous two quarters. As shown in Table 4.15 the system throughput cannot meet the market demand with the machine capacity reconfiguration. There are eight bottleneck machines in this quarter, two of which have extra over 80% capacity requirement. With so many bottleneck machines in the system, process scheduling has difficulty to feed parts to all machines just in time. In this situation, the above reconfiguration only increases the machines to near 100% capacity requirement. As long as waiting occurs, there are no enough time to complete the whole processing hence the system cannot meet the market demand.

Table 4.13: Required capacity for bottleneck machine in Q4.

Machine	Required extra capacity	Reconfiguration setting
A	0.50%	0.5%
B	19.96%	20%
C	88.42%	100%
E	83.33%	100%
G	11.50%	12%
H	32.31%	32%
I	59.58%	60%
J	5.25%	5%

Table 4.14: Machine utilisation before and after reconfiguration.

	Before		After	
Machine	**% Busy**	**% Idle**	**%Busy**	**%Idle**
A	65.31	34.69	96.83	3.17
B	79.99	20.01	96.03	3.97
C	100.00	0.00	96.08	3.92
D	88.70	11.30	93.39	6.61
E	100.00	0.00	100.00	0.00
F	76.03	23.97	76.03	23.97
G	97.68	2.32	99.46	0.54
H	95.64	4.36	94.12	5.88
I	98.68	1.32	71.69	28.31
J	17.79	82.21	45.12	54.88
K	43.33	56.67	41.02	58.98
L	22.13	77.88	22.13	77.88

Table 4.15: System throughput before and after reconfiguration.

		P1	P2	P3	P4	P5
Throughput	**Before**	273	714	676	195	234
	After	1076	1110	676	195	234
Market Demand		1599	1404	676	195	234

V. Quarter five results and analysis

As the market demand starts to decline in Q5, there are only three bottleneck machines in the system. Moreover, the production of five products except product P1 meets the market demand. The reconfiguration gives extra capacity and the system produces extra 473 unit P1 after reconfiguration.

Table 4.16: Required capacity for bottleneck machine in Q5.

Machine	Required extra capacity	Reconfiguration setting
C	30.58%	31%
E	21.25%	21%
I	24.67%	25%

Table 4.17: Machine utilisation before and after reconfiguration.

	Before		After	
Machine	**% Busy**	**% Idle**	**%Busy**	**%Idle**
A	61.93	38.07	70.33	29.67
B	76.54	23.46	89.16	10.84
C	100.00	0.00	99.50	0.50
D	66.28	33.72	66.28	33.72
E	100.00	0.00	100.00	0.00
F	53.61	46.39	53.61	46.39
G	74.76	25.24	74.76	25.24
H	86.78	13.22	86.78	13.22
I	89.66	10.34	85.11	14.89
J	41.28	58.72	46.20	53.80
K	29.17	70.83	29.17	70.83
L	16.58	83.42	16.58	83.42

Table 4.18: System throughput before and after reconfiguration.

		P1	P2	P3	P4	P5
Throughput	**Before**	434	845	455	65	247
	After	906	845	455	65	247
Market Demand		1365	845	455	65	247

The one step reconfiguration experiments over the five quarters proved the simulation method for reconfiguration investigation is reasonable and satisfactory. The results give an overview of the system performance and benefit the further reconfiguration study. The major problem in the one step reconfiguration is that even if machine capacities are reconfigured to meet the demand requirment, there is still a gap between the demand and the system throughput. As explained before, it is due to the complexity of processing scheduling. From the results of the machine utilisation, it was observed that even if all the machine utilisations are under 100%, the system output is still not guaranteed to meet the market demand. This points out that machine utilisation is not the accurate indication for bottleneck machine. Machine capacity improvement problem and the process scheduling will be further discussed in Sections 4.7 and 4.8.

4.7 Extra machine capacity can reduce the overall output

It was illustrated in the reconfiguration results of Q1 in Figure 4.10 that increasing machine capacity over the demand requirement can reduce the overall throughput. The one step reconfiguration based on the capacity requirement calculated by TOC theory does not solve the production problem for all five quarters. In which throughput does increase, it is still below the market demand. In the current context, the result indicates that this is due to the scheduling of the production system.

In Figure 4.10, production throughput meets the market demand with 23% to 26% improvement of Machine F and drops again upon the further improvement. The total throughput of the system should continuously improve or stay the same as the market demand when Machine F's cycle time is improving. However, the simulation results show that the total output will go down with extra capacity on one machine. In the other quarters, the system throughput cannot meet the market demand even if the reconfiguration level is set to meet the requirement. This result is unexpected from the TOC point of view.

In order to discuss the reason for such system behaviour, a small simple WITNESS model was built. A screen-print for a simple three machine two product (2P/3M) models is shown in Figure 4.13:

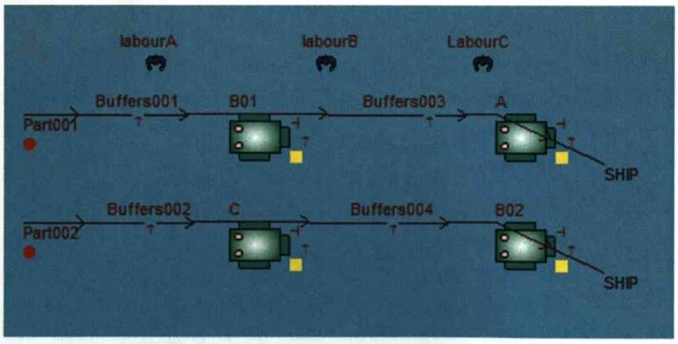


Figure 4.13: Simple 2P/3M model.

This 2P/3M model has three machines working on two different parts and finally output two products. No assembly is required. Part001 is processed on Machine B and Machine A while Part002 is processed through Machine C and Machine B. The two parts turn into Product P1 and P2 after completion of two processes. The two parts share Machine B during the production. Product P2 has priority since it makes more profit. Tables 4.19 and 4.20 show other processing data for this model.

Table 4.19: Machine cycle time for the 2P/3M model.

Machine cycle time (minutes)			
A	B01 (for part001)	B02 (for part002)	C
20	10	10	30

Table 4.20: Market demand for two products.

Product demand (per 300 minutes)	
Product P1	Product P2
15	17

It is quite straightforward to observe that Machine C is the bottleneck of the system. There is only enough time for Machine C to process 10 part002 for Product 2. The system can just about to produce enough P1 but cannot meet the demand for P2. Improving the capacity of Machine C will increase production output. The simulation model presents the system output while changing Machine C cycle time from 30 to 5 minutes with 1 minute step size. The production output is given in Figure 4.14.

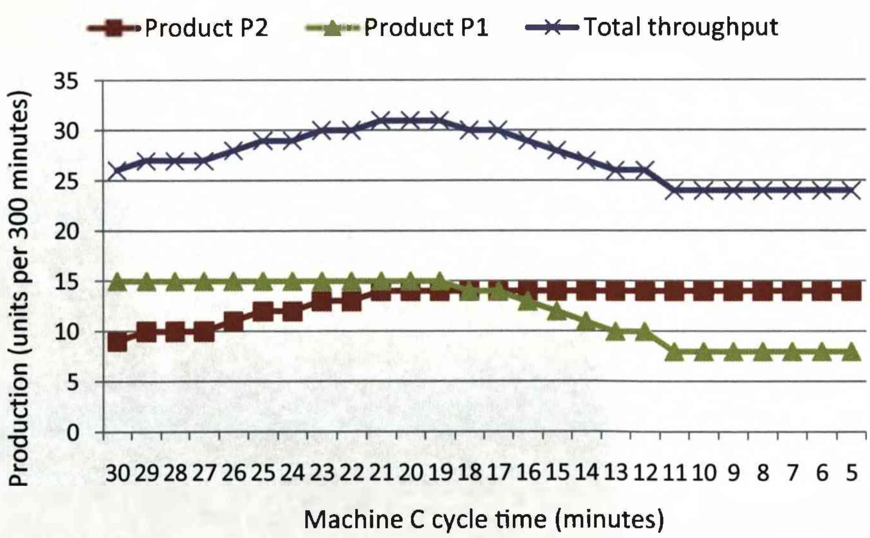


Figure 4.14: Production output changes with improvement of machine C.

Figure 4.14 shows when the bottleneck machine C becomes faster, the system's output will increase at the beginning. However, when the improvement is over 20 minutes per part, the overall production starts to decline as the output of P1 is dropping. This is the same scenario found in reconfiguration experiment for the 5P/12M system at Q1. However, in this situation it is a clearer and easier to explain.

When Machine C processes at the speed of 20 minutes per part, Machine B still has enough time to process one part001 before going back to process part002. Machine C is the precedent machine and feeds part002 to Machine B. As P2 has priority, it will only process part001 and supply them to Machine A when Machine B does not have part002 to process. Nevertheless, with the improvement of Machine C's capacity, Machine B has to process part002 continuously according to the priority rule. Under this scheduling rule, Machine B will have to complete all processes on part002 before starting to process part001. In this manufacturing case, Machine A is idle at the start when Machine B is working on the part002. However, when Machine B completes the process for part002, there is no enough time for Machine A to process all the part001 required for the demand period.

From the above analysis, it is obvious that the production decline is caused by the system scheduling. If the scheduling of Machine B can ignore the priority and set to produce P1 first, the system output will not be affected by the reconfiguration. Unfortunately, it is very easy to observe the fact in this simple system but very difficult or impossible to schedule when the production system is complex. This is also the reason why there are so many different scheduling rules in the literature and industry.

From the simulation results it is observed that the changes of production output happen when a machine improvement changes around the constraints threshold calculated by the TOC theory and when it is no longer a bottleneck machine. The improvement of a bottleneck machine has dramatic influence on other machines performance. A closer look of the system, one will find that the appearance of a new bottleneck machine will be the new obstacle of overall improvement after the current bottleneck has been removed.

In the 5P/12M model, a comparison experiment can further demonstrate the situation. The first scenario sets the improvement of machine F to 24% and the second one is set to 35% both in Q1. The simulation runs for only 2000 minutes in both scenarios in order to observe the effect of Machine F's improvement on other machines' utilisations. The results are shown in Table 4.21.

Machine F is focusing on processing parts needed for product P5 as it has the highest priority. As expected, with the improvement of machine F, the processing time for

P5 is reduced and there are 5 units more P5 produced in the same simulation time period. Consequently, the system produces a lot less product P1 and P2. The explanation for this production change is similar to the simple 2P/3M model and also shows in the changes of machine utilisations. Except machine F, nearly all other machine utilisations are reduced. It is because when increasing machine F capacity, one or more subsequent machines have to work on the extra parts feed by Machine F. They would not have enough time to process parts for less priority products which other machines are waiting for. The result of this chain effect is the increase of downstream machine idle time and the decline of machine utilisations.

Table 3.21: Comparison for different level improvement of machine F.

	Scenario 1	**Scenario 2**
Improvement	**24%**	**35%**
Production:		
Product1	36	21
Product2	39	32
Product3	2	3
Product4	0	0
Product5	66	71
Machine Utilisations %:		
A	48.77	40.17
B	100	99.98
C	86.21	72.01
D	68.16	66.43
E	67.03	52.17
F	100	100
G	74.73	72.62
H	98.83	98.22
I	65.26	53.94
J	43.66	31.26
K	2.11	3.16
L	58.79	62.79

In conclusion, the one step reconfiguration does not satisfy the demand. Increasing the machine capacity only will not solve the overall production problem. The production schedule will have to be rearranged for the new configuration and the optimisation of the production output.

4.8 Scheduling rules

4.8.1 TOC based scheduling and shortest processing time

Theory of Constraint is not only a business philosophy but also a problem-solving paradigm that establishes a clear distinction between two pivotal aspects of a problem: (1) a precise definition of the constraints that define the problem to be solved; (2) the algorithms and heuristics enabling the selection of decisions to solve the problem. It is because of these capabilities that TOC is increasingly being employed as a problem solving path for scheduling problems. Literally speaking, the TOC based scheduling or constraints based scheduling is statistically validated and performs high-quality results (Baptiste, 2001). Specifically the concept of drum-buffer-rope and buffer management is applicable to scheduling decision-making.

The scheduling rule applied in the 5P/12M simulation model is a TOC based scheduling method. The scheduling decision is made upon the priority of products calculated based on Theory of Constraint. The higher the profit on bottleneck machine time is, the higher the priority is. This scheduling rule has been discussed in Chapter three in details.

As concluded in the previous section, scheduling is the reason beyond system constraints that prevents the production to meet the market demand. There are quite a few scheduling rules available in the literature although none of them is out perform others in all measurement methods. One is good at increasing the production quantity may not be the best way to maximum the profit; one is good at reducing the WIP could result a poor system throughput.

In order to test the impact of scheduling rules on the model studied, it is very important to apply a second scheduling rule to the reconfiguration investigation. Shortest Processing Time is one of the most popular scheduling rules in the literature, due to the fact it is very easy to use and there is less calculation and control on the implementation to various manufacturing systems. It is evidenced that the Shortest Processing Time produces a system performance which is above the average outcome on the changing of scheduling rules (Ferrell, Sale, Sams, & Yellamraju, 2000). In addition, since there is no adequate aspect of system data in the manufacturing model, the range of scheduling rules which can be applied is highly limited. Therefore, scheduling rule shortest processing time has been chosen to test in the 5P/12M model reconfiguration.

4.8.2 Impact of scheduling rules on system output

A group of simulation experiments was set to examine the impact of two different scheduling rules. Generally speaking, scheduling rules do not affect the system output in a certain period of time if the system can well meet the market demand and machines all have excess capacity. Only when the system cannot meet the market demand, the divergence of employing the different scheduling rules becomes obvious and important. The manufacturing system studied in this research has bottleneck machine or multiple bottleneck machines; in other word the system cannot meet the demand in all five quarters. Scheduling rule will certainly make a difference on this model.

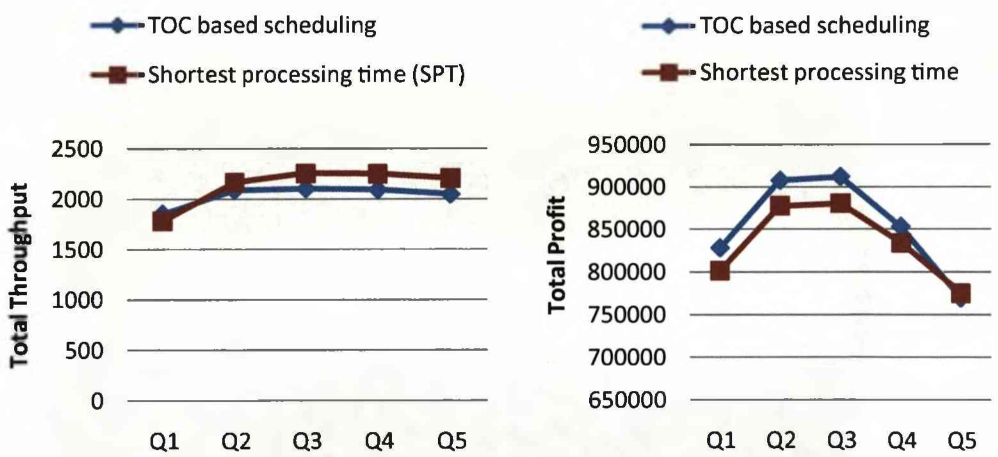


Figure 4.15: Throughput and profit output for two scheduling rules.

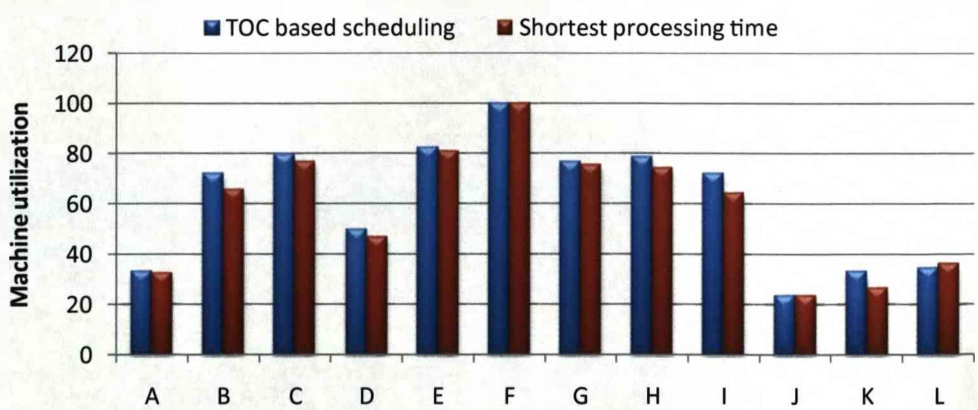
The comparison of the system output under the two scheduling rules is shown in Figure 4.15. The experiments are carried out with original configurations when the system is under production over all five quarters. The divergence of the TOC based scheduling rule and the Shortest Processing Time is illustrated on both the system throughput and total profit in Figure 4.15. The diagram clearly illustrates that the two scheduling rules are established from different objectives hence result differently on the system outcome. The TOC based scheduling prioritises the profit per bottleneck machine time, therefore it results more profitable output than the Shortest Processing Time. On the contrary, Shortest Processing Time neglects the profit but values the quick processing products, hence the total production throughput are higher in all five quarters. The performance of the two scheduling rules after machine reconfigurations continuously shows the same preference. The results are demonstrated in Appendix A.

4.8.3 Machine utilisations change with different scheduling rules

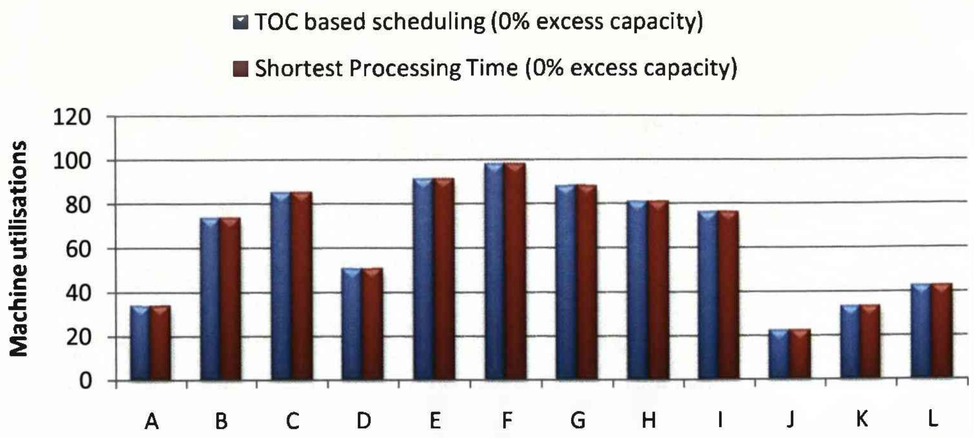
The change of scheduling rules not only affects the system output but also has influence on the machine utilisations. The following experiments are carried out for the 5P/12M model at Q1 demand in original and three reconfiguration settings.

Table 4.22: Machine utilisation change at Q1 demand.

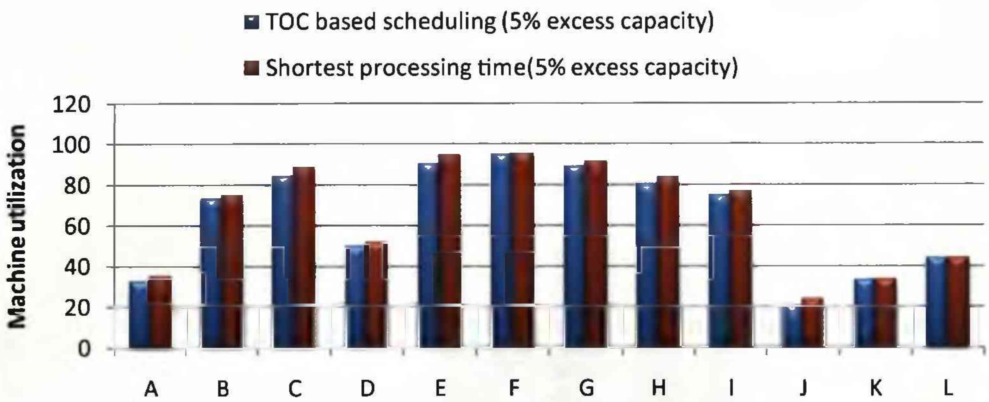
	TOC based scheduling				Shortest Processing Time			
Capacity	*original*	*100%*	*105%*	*110%*	*original*	*100%*	*105%*	*110%*
A	33.37	33.75	32.6	32.87	32.71	33.75	35.23	35.94
B	72.03	74.04	73.49	73.38	66.13	74.04	75.15	76.66
C	79.85	85.3	84.19	84.67	77.14	85.3	88.57	90.45
D	49.91	50.47	49.89	50.52	46.69	50.47	51.78	52.8
E	82.34	91.71	90.04	87.44	81.13	91.71	94.63	92.62
F	100	98.39	95.15	92.35	100	98.39	95.41	94.09
G	76.88	88.05	89.18	88.33	75.45	88.05	91.23	91.49
H	78.97	81.04	80.74	82.69	74.43	81.04	83.64	85.34
I	72.08	76.3	75.11	72.72	64.54	76.3	77.08	78.64
J	23.75	22.44	20.7	20.48	23.75	22.44	23.75	24.2
K	33.33	33.33	33.33	33.33	26.78	33.33	33.33	33.96
L	34.75	42.67	43.94	44.42	36.29	42.67	44.11	45.25



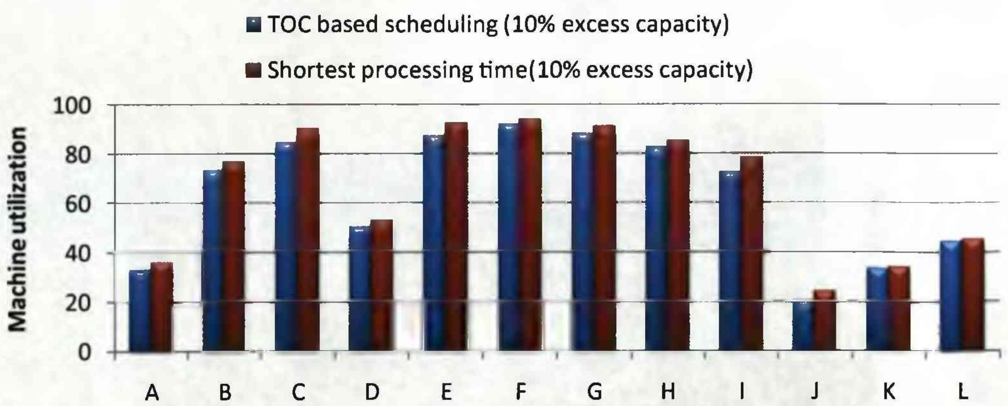
(a)



(b)



(c)



(d)

Figure 4.16: Machine utilisations changes with the scheduling rules.

Experiments to test the impact of scheduling rules on machine utilisations are carried out in Q1 demand first and there is only one bottleneck machine. The results shown in Table 4.22 and Figure 4.16 are under four system configurations which are the original configurations and three configurations after the improvement of Machine F's capacity. The improvements are calculated based on the TOC theory at 100% satisfying the capacity requirement and 5% or 10% above the capacity requirement. It is because at the previous experiment, even the available capacity reconfigure to 100% requirement the system output still does not meet the demand. Excess capacity could be a solution to compensate the scheduling loss.

It can be observed from Table 4.22, the utilisation of Machine F is gradually reducing while they are increasing for other machines with the raise of system throughput. The comparison of TOC based scheduling and the Shortest processing time in Figure 4.16 shows an interesting result. Machine utilisations are higher in TOC based scheduling than the Shortest processing time before the reconfigurations while they are equal on 100% reconfiguration and lower with excess capacity on Machine F after reconfigurations. The manufacturing system meets the market demand after Machine F's reconfiguration at 24% improvement. The system throughput equals market demand hence the machine utilisations are the same for both scheduling rules. Before the reconfigurations, the TOC based scheduling has higher throughput than the SPT shown in Figure 4.15. After reconfiguration the system throughput is higher with SPT rule hence the machine utilisations are higher.

Parallel experiments are carried out for the other four quarters and the simulation results are gathered in Appendix A. The results obtained during these experiments confirmed the aforementioned conclusions.

Chapter 5
Optimisation for the Reconfiguration of Manufacturing System

5.1 Introduction

The application of metaheuristics to combinatorial optimisation problems is a rapidly growing field of research. This is due to the importance of combinatorial optimisation problems for the scientific as well as the industrial world. However, even with high speed, state-of-the-art computing resources, an exhaustive search can take a prohibitive amount of time for manufacturing optimisation. To circumvent this problem, optimisation algorithms which can be used with computer simulation models are more efficient for identifying optimal/near-optimal manufacturing process designs.

Six optimisation algorithms embedded in WITNES software are used after a study of combinatorial optimisation problem and metaheuristic algorithms. The method of finding an optimal work in progress value using optimisations was explained and illustrated in Section 5.4. All six WITNESS optimisation algorithms are used to search the reconfiguration of the manufacturing system in Section 5.5 and simulated annealing and hill climbing algorithm are compared in Section 5.6. The impact of parameter step size is analysed in the last section.

5.2 Combinatorial Optimisation problem

The term optimisation refers to a search process for a minimum or maximum result of a real function. As a branch of optimisation, the Combinatorial Optimisation (CO) deals with discrete problems where the goal is to find the best possible, feasible and discrete solution. In other words, we are looking for an optimum from a finite - or possibly countable infinite- set (Steiglitz, 1982).

5.2.1 Description of the combinatorial optimisation problem

A Combinatorial Optimisation problem P = (S, *f*) can be defined by:

- *a set of variables* $X = \{x_1, x_2, \ldots, x_n\}$;

- *variable domains $D_1, \dots, D_n$;*
- *$v_1, \dots, v_n$ variables within the domains;*
- *constraints among variables;*
- *an objective function f to be minimized, where $f: D_1 \times \dots \times D_n \rightarrow \mathbb{R}^+$* positive real number;

The set of all possible feasible assignments is

$S = \{ s = \{(x_1, v_1), \dots, (x_n, v_n)\} \mid v_i \in D_i,$ *s satisfies all the constraints*$\}$.

S is usually called a search (or solution) space, as each element of the set can be seen as a candidate solution. To solve a combinatorial optimisation problem one has to find a solution $s^* \in S$ with minimum objective function value, that is, $f(s^*) \leq f(s)\ \forall s \in S$. s^* is called a globally optimal solution of (S, f) and the set $S^* \subseteq S$ is called the set of globally optimal solutions (Blum & Roli, 2003). Although it is restricted to minimization problems here, all results can be extended easily for maximization problems.

5.2.2 Metaheuristics Optimisation algorithms

The Travelling Salesman Problem, the Minimum Spanning Tree problem and Timetabling and Scheduling problem are typical examples of combinatorial optimisation problems. For most optimisation problems, there is no known algorithm that solves all instances quickly. Furthermore, it is unlikely for such an algorithm to be discovered. The combinatorial optimisation problems are known as NP-hard. However, having a problem classified to be NP-hard still require a solution or near optimum solution. Many algorithms such as genetic algorithm, ant colony and simulated annealing to tackle the problems have been developed.

Algorithms searching the complete combinations are guaranteed to find for every finite size instance of a combinatorial optimisation problem an optimal solution in bounded time (Wolsey, 1988). A good result is guaranteed but complete methods might need exponential computation time which leads to excessive amount of time for practical purposes. Consequently, approximate methods to solve CO problems have been developed extensively over the last few decades.

Metaheuristic literally means to find a solution in an upper level as the word is derived from the composition of two Greek words. *Heuristic* is derived from the verb *heuriskein* which means 'to find', while the suffix *meta* means 'beyond, in an upper level'. The term is used to categorise the kind of approximate algorithms which try to combine basic heuristic methods in higher level frameworks aimed at exploring a search space efficiently and effectively. Ant Colony Optimisation (ACO), Genetic Algorithms (GA), Iterated Local Search (ILS), Simulated Annealing (SA), and Tabu Search (TS) and so on are commonly accepted as metaheuristic algorithms. The following are three definitions of metaheuristic in the research field.

A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions (Laporte, 1996).

A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method (S. Voß, 1999).

A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimisation problems with relatively few modifications to make them adapted to a specific problem (www.metaheuristics.net, 2000).

Metaheuristics are a general class of heuristics for solving NP hard problems. They are sometimes considered as intelligent heuristic search, which can avoid the local optimality and incorporate various strategies inspired from natural behaviours of species, mathematical reasoning, physical science, nervous systems and statistical mechanics.

5.3 Optimisation algorithms in WITNESS

5.3.1 WITNESS Optimizer

The WITNESS Optimizer module can be integrated seamlessly with the standard WITNESS system. Using the latest in sophisticated mathematical technique, it offers a fully customisable optimisation platform for searching the optimal result by identifying and changing the model parameters. The powerful facility is dedicated to the automatic testing of simulation models to indentify the rigorous what-if experimentation in order to explore alternative manufacturing scenarios such as processing times, quantity, staff numbers, and queue sizes and so on. WITNESS Optimizer not only searches for the optimum solution but also is capable of completing the analysis quickly. Using intelligent algorithms, which measure and react to the success of each analysis path, the best solution is often found by trying less than 1% of all possible combinations (www.lanner.com).

5.3.2 Optimizer algorithms

Six different optimisation methods are provided in the WITNESS Optimizer, ranging from simply running all combinations through to the more complex and intelligent metaheuristic algorithms. These include:

- **All combinations** - this option should indeed be chosen if time allows as it covers all possible combinations. However, it guarantees that the best result within the parameters range will be found. An estimate of the time to be taken can be obtained after a sample run.
- **Random solutions** - to enable an appreciation of the shape of the solution space by generates random combinations results.
- **Min/Mid/Max** - tests the extremes and mid points of all parameter settings. Covers all options for non-range parameters.
- **Hill Climbing** - a simple algorithm. This method iteratively generates a single neighbour, which is accepted only if it is of higher quality. The neighbourhood function selects a parameter at random; if the parameter is continuous then it is either increased or decreased at random. Non-continuous parameters are set to a value picked at random from their valid range.

- **Adaptive Thermostatistical Simulated Annealing** - The main algorithm, a variant of simulated annealing with extra adaptive nature. Includes some elements of tabu search. Developed by Lanner in conjunction with optimisation experts specifically to tackle simulation experimentation.
- **Six Sigma Algorithms** - based on the Simulated Annealing method. With this method the level of changes to a model can be limited for the purpose of identifying the best options for process improvement.

5.3.3 Hill Climbing algorithm

Hill-climbing is a local search technique, which moves from one solution to another one in the neighbourhood. If the quality of the new solution is better than the previous one, this move is accepted and the search continues from the new location. If the neighbouring state does not result in an improvement, the move is rejected and the search continues from the current state. The main disadvantage of this method is that the search process might get trapped in a local minimum, which is not equal to the global one. A useful variation on simple hill climbing considers a series of moves from the current state and selects the best one as the next state. This method is known as gradient search or steepest-ascent hill-climbing.

In order to overcome the main disadvantage of local search algorithms such as hill-climbing, whose weakness lies in the inability to escape from local minima, more sophisticated heuristic search strategies are designed to avoid such a situation. This implies the temporary acceptance of a state of lower quality. Hence meta-heuristic algorithms can be considered to some extent as local search strategies; however, they include a means to escape from local minima.

5.3.4 Adaptive Thermostatistical Simulated Annealing

A unique algorithm known as Adaptive Thermostatistical Simulated Annealing is the main optimisation algorithm in the WITNESS Optimizer which offers fast and effective performance in searching for optimal solutions (www.lanner.com).

The WITNESS Optimizer uses Simulated Annealing and Tabu search with elements of reactive thermo-statistical search using an adaptive cooling schedule. Simulated Annealing is based on the concept of local search and designed to reduce the risk of becoming trapped in local optima as worse solution can be accepted as well. At each stage of the search, a neighbour of the current solution is generated and either accepted as the new solution or rejected. This acceptance process is initially random, but becomes increasingly dependent on solution quality as time goes on. The temperature controls the degree of randomness present within the search and is modulated by a predetermined cooling schedule.

The Simulated Annealing algorithm has the following implementation process:

```
generate an initial solution s_0,
        select an initial temperature T_0 > 0
        set current temperature T = T_0
        select a cooling schedule parameter α < 1
        select a number of iterations n to spend at each temperature
        repeat
                repeat
                        generate a neighbour s of s_0
                        δ = quality(s_0) - quality(s)
                        if δ < 0 then s_0 = s
                        else
begin
generate random x uniformly in the range (0,1)
        if x < exp(-δ/t) then s_0 = s
end
                until number of iterations performed at this temperature step = n
                update temperature by T = αT
        until stopping condition is met
```

The initial temperature and cooling schedule used within Simulated Annealing is highly problem-dependent and can vary widely. The choice of cooling schedule has a considerable impact upon the quality of the resulting solution. Parameter optimisation is therefore essential if high quality results are to be obtained. The initial temperature determines the degree of randomness that is initially present in the search. Higher initial temperatures will introduce a greater degree of randomness within the search.

In the algorithm described above, the cooling schedule parameter (α) controls the rate at which the temperature is reduced; large values will produce slow cooling schedules. The parameter value must always be less than one, since it must reduce the temperature, and must be greater than zero, since the temperature must not become negative, nor reach zero in a single step. In practice, values of the parameter between 0.7 and 0.95 tend to be used. The length of the temperature step also controls the cooling rate. Longer temperature steps will produce slower cooling rate if the parameter α remains fixed. The recommended number of temperature steps is approximately 25. It is also a good practice for the cooling schedule parameter and temperature step length to be set so that the final temperature is approximately 10% of the initial temperature (www.lanner.com).

The WITNESS Optimizer includes options for the automatic setting of the above temperature parameters. It also includes an optional adaptive cooling schedule that is based on expert advice on practical parameter optimisation within a Simulated Annealing algorithm. Expert users can modify these settings as required.

Reactive Thermostatistical Search (RTS) is a new technique proposed by Chardaire et. al in 1995 (Chardaire, 1995). It is also used in the WITNESS Optimizer and RTS incorporates elements of Tabu search into the simulated annealing process. It incorporates the learning process of Tabu search and studies its experience of the problem domain and modifying its search strategy accordingly. Tabu search, like simulated annealing, is based upon the local search paradigm and in its basic form searches by iteratively moving from a single current solution to one of its neighbours, until some pre-set conditions are met. However, rather than introducing randomness into the search to avoid being trapped in local optima, Tabu search makes use of a collection of rules to determine the nature of its search. This rule list allows Tabu search to operate in a very sophisticated manner, taking into account factors such as past experience or problem domain knowledge when conducting the search. Reactive thermostatistical search monitors the performance of each of the parameters and adapts accordingly. The search gives bias towards parameters which, when modified, give solutions that are accepted as replacements for the current solution.

The parameter bias is implemented within reactive thermostatistical search using an adaptive neighbourhood. Rather than selecting a parameter at random to modify in the generation of a neighbour, the technique selects each parameter with a probability based on its past performance over a number of iterations.

The term Simulated Annealing used in the following context is referred as the Adaptive Thermostatistical Simulated Annealing embedded in WITNESS Optimizer.

5.3.5 Optimisation process in Optimizer

1. Set up one or a few objective functions

There are two main functions set up in the WITNESS model. They are objectives of the optimisation process.

Total_throughput

Return *throughput_P1+ throughput_P2+ throughput_P3+ throughput_P4+ throughput_P5*

Profit

RETURN $throughput\_P1 * (P_{p1} - P_1 - P_3 - P_{10}) + throughput\_P2 * (P_{p2} - P_1 - P_6 - P_7 - P_{14}) + throughput\_P3 * (P_{p3} - P_3 - P_9 - P_{17} - P_{19} - P_{20}) + throughput\_P4 * (P_{p4} - P_4 - P_8 - P_{13} - P_{16} - P_{18}) + throughput\_P5 * (P_{p5} - P_2 - P_5 - P_8 - P_{11} - P_{12} - P_{15})$

Where P_1 to P_{20} are the cost of twenty raw materials and P_{p1} to P_{p5} are the sale prices of the five products. Here the operation cost is not counted in as it can be regarded as a constant hence it will not affect the result of profit change.

In addition, five small individual functions return the throughput value for product P1 to product P5 are set to monitor the changes of the throughput amount for each product.

2. Optimizer settings

The Optimizer command is located in the WITNESS model menu and brings up a central dialog to setup the optimisation. Without an objective function, an error message will stop the Optimizer program.

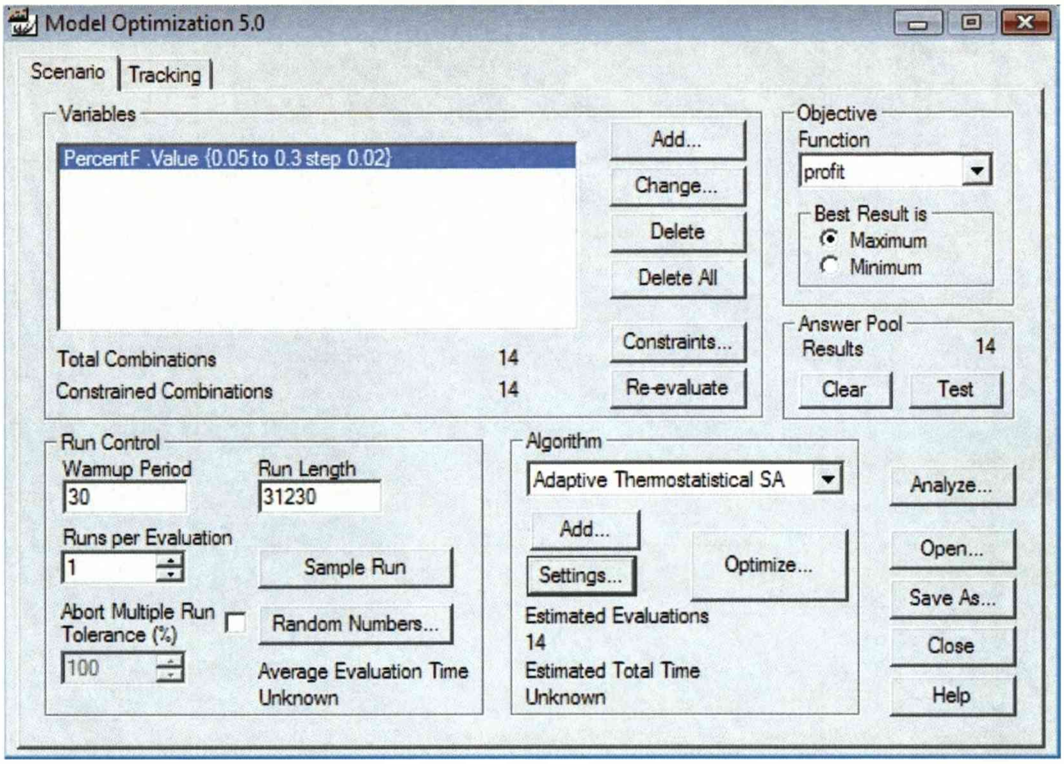


Figure 5.1: Optimisation experiments dialog.

The combinations of optimisation are decided by user chosen variables. Figure 5.1 is an example of optimisation experiment setting. The user can add the value of PercentF as the experiment parameter varying from 0.05 to 0.3 with a step size 0.02. Therefore, the total combinations in the optimisation have 14 runs. Other constraints can be entered as well, for instance, relationship between parameters. This condition can be entered as a constraint. There is only one objective function per experiment which can be chosen from the drop down list.

Other options in the Run Control section, such as run length and number of replications should then be set in line with the simulation period. As there is no random number used in this model, there is no need to run multiple times on each evaluation.

Finally, the algorithm to be used to carry out the optimisation process can be chosen from the six built-in algorithms. WITNESS allows the capability to add further optimisation algorithms, however this research does not utilise this feature as the built-in algorithms are generally adopted in manufacturing research.

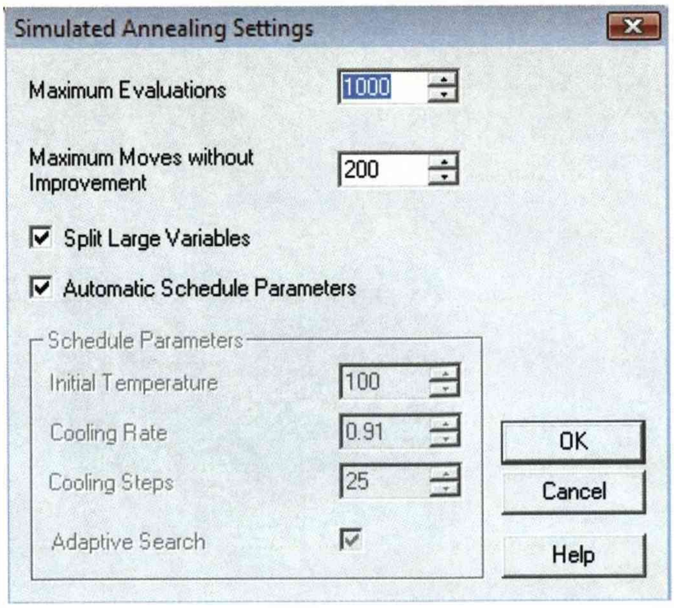


Figure 5.2: Simulated Annealing settings.

To search effectively, automatic settings can be chosen which will adapt the algorithms to the model through an analysis as it runs. Depending on the number of total combination, WITNESS assigns the number of maximum evaluations and maximum moves without improvement accordingly. However, one can increase that if a longer simulation time is allowed. It is not recommended to shorten evaluations from the automatic settings.

If the user is very familiar with the use of modern heuristic algorithms, a number of key parameters such as initial temperature, cooling rate and cooling steps can be adjusted manually. This may provide better solutions as it may tailor the settings to the users' knowledge about the problem under investigation.

3. Gathering experiment results

A number of reports are produced as the algorithms run, which include tables and graphs of the results. The reports include input value settings, the objective result, variance and confidence tables and graphs for replicated runs. Figures 5.3, 5.4 and 5.5 are given as examples.

Figure 5.5 is an optimisation object graph generated in WITNESS Optimizer in its standard format. Each combination in the optimisation search is recorded as an evaluation in the x axis. In this optimisation search, 96 combinations are tested. The *y* axis is the result of object function returned from WITNESS model testing all these parameter combinations. In Figure 5.5, the result records the value of total profit under different combination of machine improvement. The black line tracks each result at the certain combination and the red line points out the best result. Figure 5.5 and subsequent figures in this chapter are presented in the format of the WITNESS Optimizer program. The axes of the figures change with the experiments performed, however, WITNESS will only show the axes as 'result' and 'evaluation'.

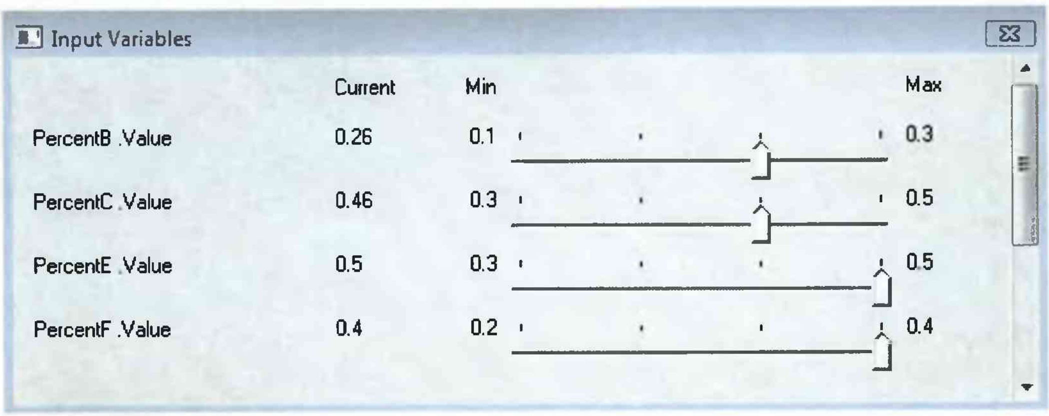
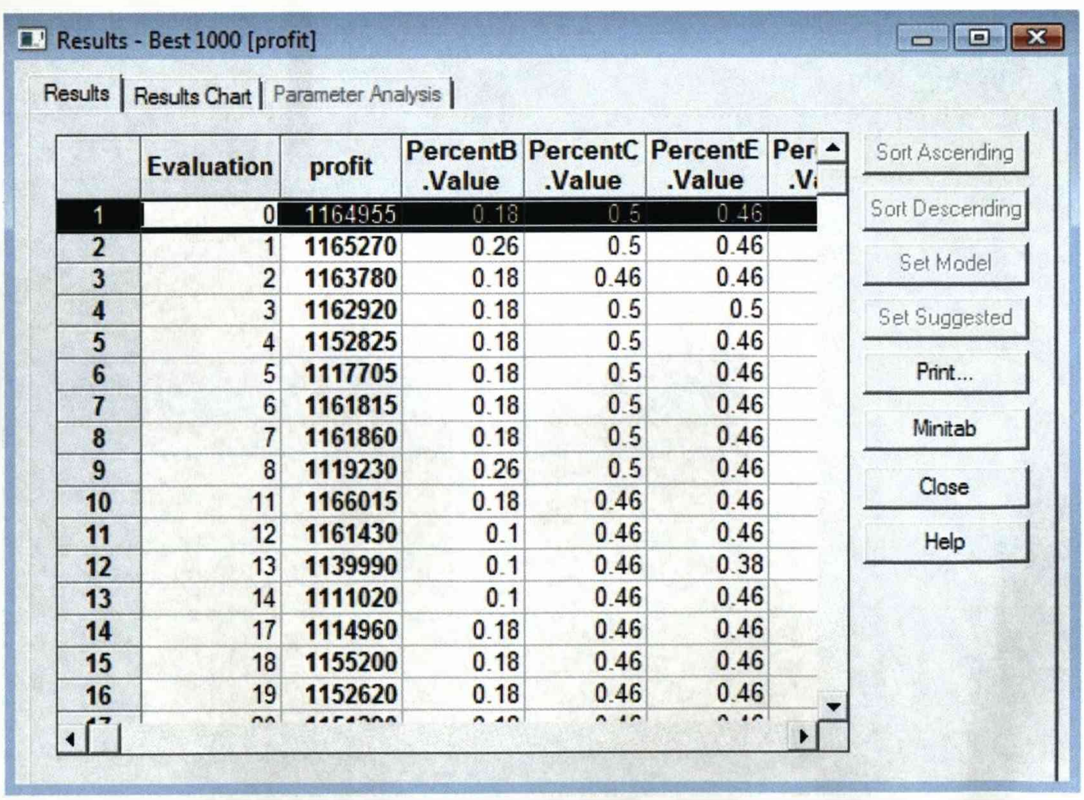


Figure 5.3: Input variables settings at optimisation.



	Evaluation	profit	PercentB .Value	PercentC .Value	PercentE .Value	Per .V
1	0	1164955	0.18	0.5	0.46	
2	1	1165270	0.26	0.5	0.46	
3	2	1163780	0.18	0.46	0.46	
4	3	1162920	0.18	0.5	0.5	
5	4	1152825	0.18	0.5	0.46	
6	5	1117705	0.18	0.5	0.46	
7	6	1161815	0.18	0.5	0.46	
8	7	1161860	0.18	0.5	0.46	
9	8	1119230	0.26	0.5	0.46	
10	11	1166015	0.18	0.46	0.46	
11	12	1161430	0.1	0.46	0.46	
12	13	1139990	0.1	0.46	0.38	
13	14	1111020	0.1	0.46	0.46	
14	17	1114960	0.18	0.46	0.46	
15	18	1155200	0.18	0.46	0.46	
16	19	1152620	0.18	0.46	0.46	

Figure 5.4: Optimisation results table.

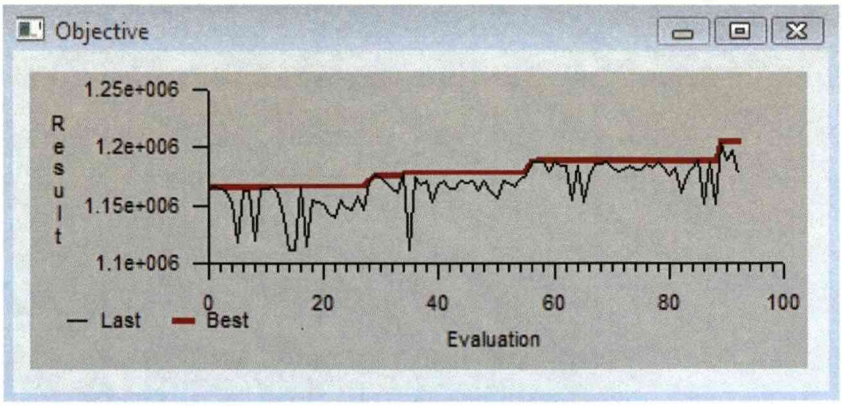


Figure 5.5: Objective result monitor during optimisation evaluations.

Algorithms are often tuned to allow a relatively accurate result to be processed within a finite amount of time and a finite amount of computing power. Therefore, there is no existing algorithm (apart from one which tries all the options) that could guarantee an optimal solution for any simulation problem. The strength of the technique implemented in this research is that it has been shown to perform very well with relatively few iterations. The creator of the software, Lanner Group claims in their website that the optimal solution could often be found by trying less than 1% of all possible combinations, this was confirmed in this research as described in Section 5.7 (www.lanner.com).

5.4 Decision on Work In Progress level using optimisation

5.4.1 Work In Progress settings in the simulation model

Production management aims to minimize work in progress. Just-in-time (JIT) production is an effort to reduce work in progress. Ideally, in a manufacturing system where JIT has been applied, the minimum WIP level can be achieved. In other words, WIP can be equal to 1. In this research, the simulation model was built with a JIT theory in mind, there is a Kanban system used in the production rules and WIP is set as a parameter to be controlled by the system manager.

NWIP( ) is a WITNESS function which returns the number of parts of products in the system. *wip* is set as an integer variable to monitor and control the system work in progress level. The condition of raw materials enter into the system is set in the first process machines of each material. For example, Machine A01 process RM1 as soon as it enters the system. In Machine A01's input dialog, the following *if* condition is used to control the WIP level:

If NWIP(RM1) < wip

PULL from RM1 out of Buffer(RM1)

ELSE

Wait

ENDIF

Where *wip* can be set in the initial action before running the simulation or it can be used as a variable in simulation optimisation. Practically, the WIP level can be different for each raw material parts. However, the WIP level is kept the same for all twenty parts in order to simplify the reconfiguration problem. Therefore, *wip* is a single integer input parameter which controls all materials WIP level and decides when a new part can be pulled into the system. This means there is *wip* amount of each raw material parts in the system waiting to be processed or assembled.

5.4.2 The best WIP level

Simulation optimisation can help to support the decision on the value of the integer parameter *wip*. Regardless the cost of keeping high WIP level, the goal to consider is the profit and throughput of the manufacturing system. Setting the total throughput to be an objective function of the optimisation search, the value of *wip* should be chosen when the max throughput is achieved. An optimisation experiment was carried out to search the best WIP level in Quarter two. The parameter *wip* varies from 1 to 1000 hence there is 1000 evaluations. Maximizing the total throughput is the objective function. The black line in Figure 5.6 shows that the throughput changes with the variation of *wip* value. It searches the maximum throughput at a very early stage.

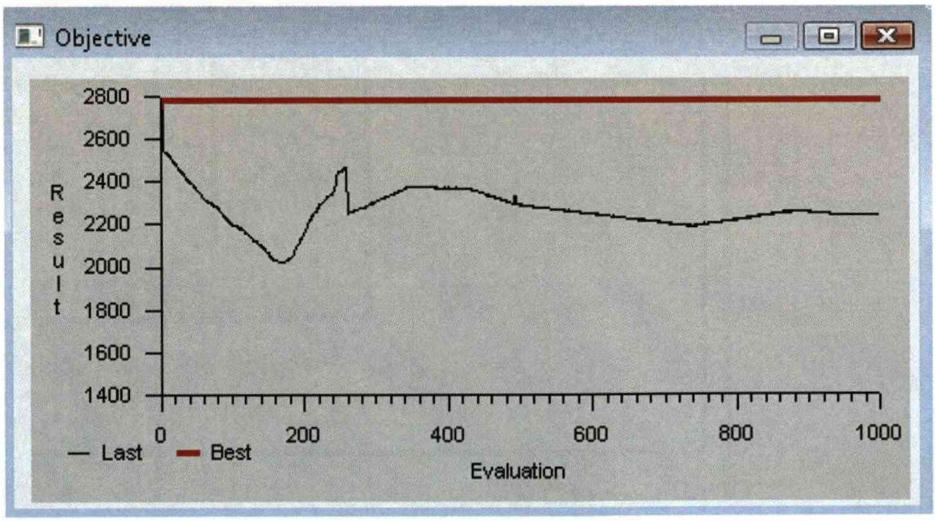


Figure 5.6: Optimisation result on variable *wip* value.

It is difficult to see the throughput changes within the first few runs of the optimisation in Figure 5.6 due to the density of the data recorded in the graph. Figure 5.7 is plotted by the data from the same experiment where *wip* varies from 1 to 10 only. It is obvious that the system throughput reaches the peak when *wip* equals to 3.

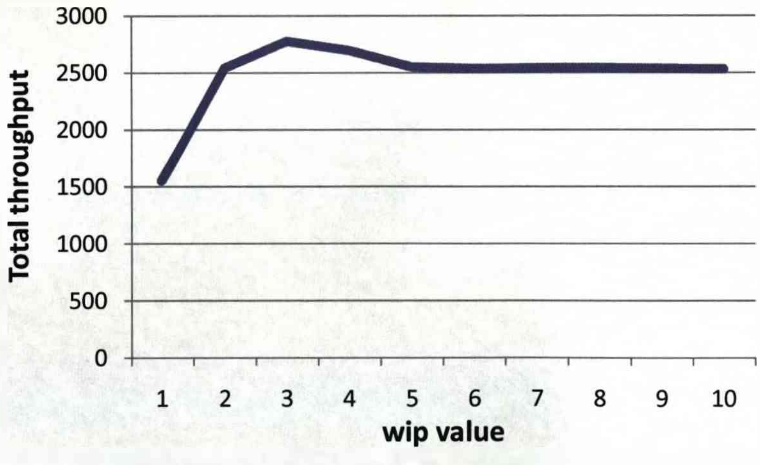


Figure 5.7: Optimisation results when *wip* is between 1 to 10.

This is a very interesting result that the system produces the maximum amount of products when the value of *wip* is 3. It is possible that the system output reaches the peak on a certain work in progress level. However, the above optimisation experiment can only prove one scenario. It seems that *wip* value equals to 3 provides the best profit after a few experiments were carried out in different demand periods. In order to further prove this result, forty optimisation experiments were carried out.

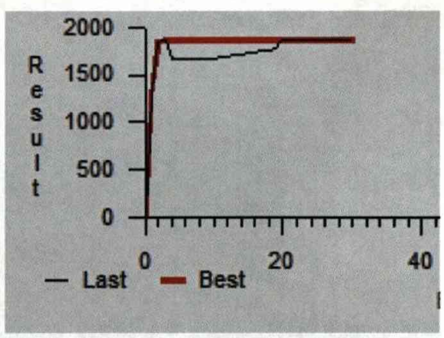
From the experience of previous optimisation, the system throughput is not sensitive to *wip* value after 10. In order to save some computation time, *wip* value in the series of experiments is set to be the following 33 selected value rather than continuous integers.

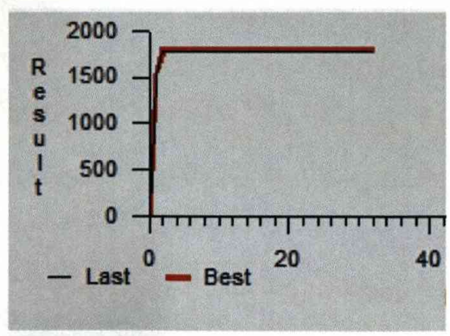
Table 5.1: Selected *wip* value for the optimisation experiments.

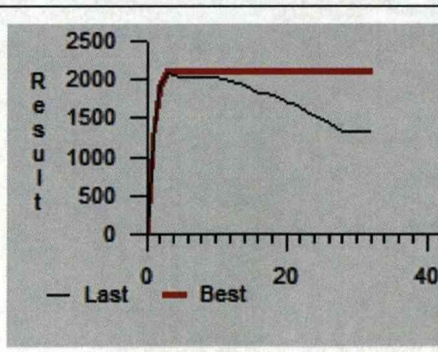
0	1	2	3	4	5	6	7	8	9
	10	20	30	40	50	60	70	80	90
	100	200	300	400	500	600	700	800	900
	1000	2000	3000	4000	5000				

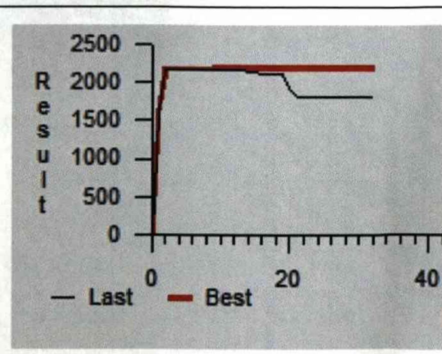
Throughput is most sensitive when wip is from 1 to 10. Therefore, the first ten runs chose the wip value with step size at 1, then increasing the step size after 10. The maximum *wip* value is set to 5000 because the maximum amount of parts which can be used in one quarter is under 5000. The first ten optimisation searches are carried out from Q1 to Q5 with two different scheduling rules.

Table 5.2: Optimisation results on different WIP level (where the common x-axis is optimisation runs and the y-axis is the system total throughput).

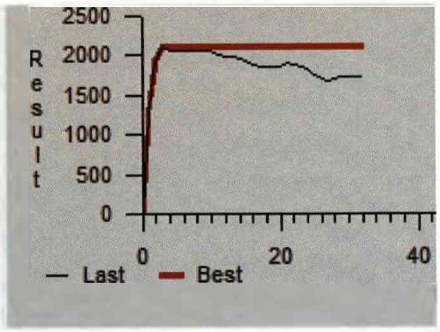
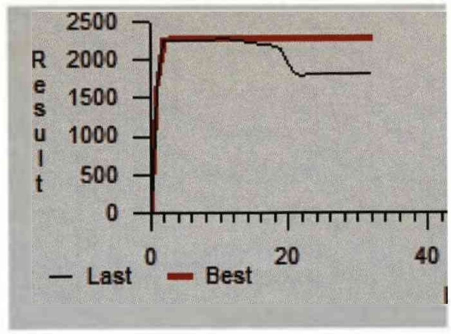
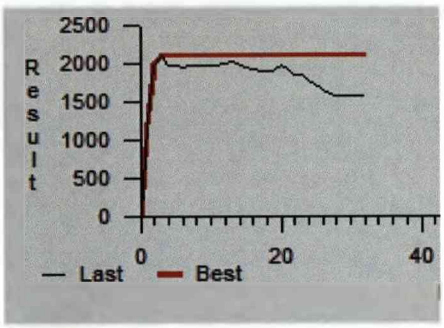
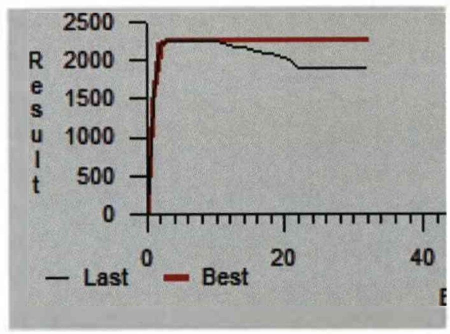
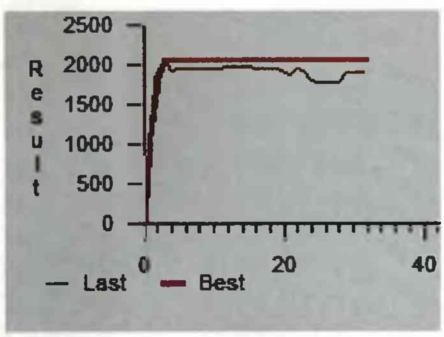
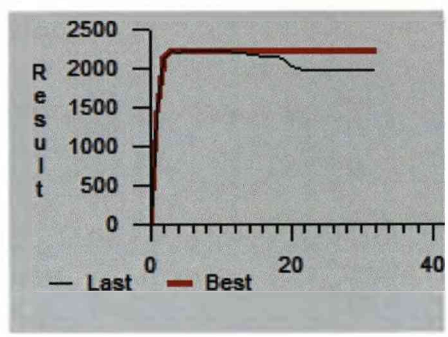
<table>
<tr><th></th><th>Highest profit on B/N machine</th><th>Shortest processing time</th></tr>
<tr><td>Quarter 1</td><td>

(a)</td><td>

(b)</td></tr>
<tr><td>Quarter 2</td><td>

(c)</td><td>

(d)</td></tr>
</table>

Quarter 3	 (e)	 (f)
Quarter 4	 (g)	 (h)
Quarter 5	 (i)	 (j)

The diagrams in the table are generated directly from WITNESS Optimizer. It is cropped from evaluation 40 as there are only 33 runs per experiment. There is no more data after evaluation 40. The x axes of the diagrams (a) to (j) in Table 5.2 are the evaluations of the 33 runs with different *wip* value set as in Table 5.1 and the result y axes are the system throughput. The black line links each total throughput of the simulation runs and the red lines mark the best result. It is very clear that the system throughput achieves the maximum when *wip* is equal to 3.

Another 30 optimisation experimental results are given in Appendix B. The experiments are carried out for different levels of system reconfigurations. Similar results are also found in such condition. Wip value equals to three gives the best system output for this particular manufacturing system.

5.5 Optimisation of the reconfiguration of manufacturing system

5.5.1 Experiments set up

The previous chapter described that manufacturing processes and production systems are complex. System reconfiguration decision cannot be made by TOC theory alone. The system output does not meet the changing market demand even after suggested machine capacity changes as shown in Section 4.6. Further optimisation experiments are required to test the system performance and assist the decision making on how and what level of machine reconfiguration. First of all, all six available optimisation algorithms in WITNESS Optimizer are tested and compared in order to provide reasons for choosing algorithms.

The following optimisation experiments are all carried out with the same scenario but with different algorithms. The experiments carried out is quarter two as it is the first demand period which has multiple bottleneck machines and the situation is in a higher level of complexity. The *wip* value is set to 3. Using TOC theory, it can be identified that there are 7 bottleneck machines in this demand period. Therefore, the 7 machine capacity improvement percentages are chosen to be the input variables throughout the experiment. The ranges of input value are chosen so that the suggested improvement values fall in and around the midpoint of the ranges. As the optimisation experiments are still formed by many discrete event simulation runs, a step size between each variable changes has to be determined. In this case, a step size of 5% is chosen for two reasons. First of all, smaller step size will make the total combinations too large to complete the experiments within a reasonable computer time. Secondly, the ranges of improvement are all 20%. Larger step size will over simplify the experiments and the results offer less accuracy.

Table 5.3: Input variables for the optimisation experiments.

	MIN	MAX	STEP
percentB	10%	30%	5%
percentC	30%	50%	5%
percentE	30%	50%	5%
percentF	20%	40%	5%
percentG	10%	30%	5%
percentH	20%	40%	5%
percentI	20%	40%	5%

With the parameter settings the total combinations are 78125.

5.5.2 Comparison and analysis of optimisation Results

Using the above experiment settings, optimisation results are summarized in Table 5.4, Figures 5.8 to 5.13 show the total profit change during each optimisation search. All combinations give the best optimisation results. However, it takes a standard Dell computer with an Intel 3.4 GHz Pentium D processor and 2GB of DDR2 RAM about 2 days and 16 hours to complete the 78125 optimisation runs. In a slightly more complex combinations, the time consumed would easily double, triple or expand exponentially so that it becomes infeasible to use this algorithm when a decision needs to be made within a limited amount of time.

Table 5.4: Comparison of optimisation results for six algorithms.

	Evaluations	Total time	Best profit	Best throughput
All combinations	78125	64:00:52	1206150	2948
Random solutions	100	00: 05: 03	1193960	2886
Min/Mid/Max	2178	01:48:19	1200355	2922
Hill Climb	265	00:13:20	1194575	2918
Simulated Annealing	310	00:21:42	1199860	2911
Six Sigma	437	00:29:25	1182575	2893

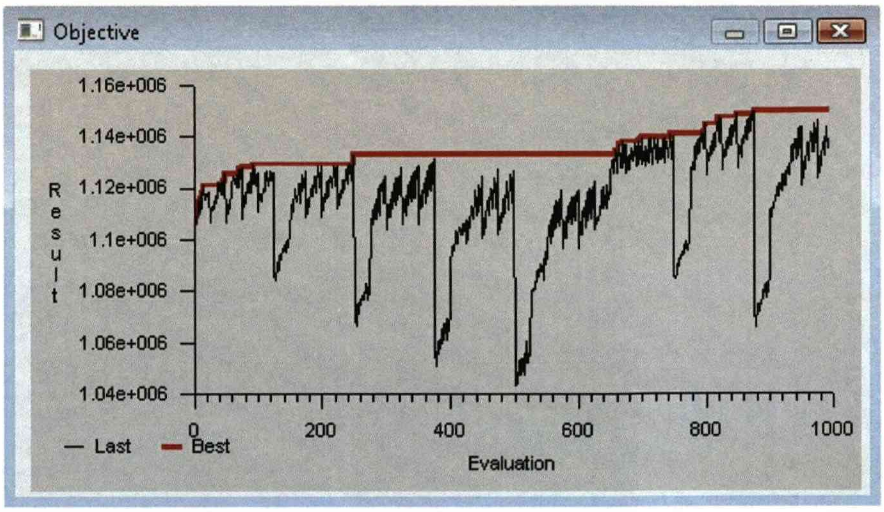


Figure 5.8: First 1000 results of All combinations algorithm.

The first 1000 optimisation results of All combinations are shown in Figure 5.8. As the maximum results in the objective window are limited to 1000 evaluations, only the first 1000 results can be displayed in the above figure. It can be observed that the trend of the plot over all evaluations has a saw wave pattern. It is because the all combination algorithm implements the experiment by changing one parameter at a time and run all the possible values concurrently.

Random solutions only used 5 minutes and 3 seconds. The length of the experiment is solely dependent on the evaluations user chooses to run. As the results are totally random, the 'random' method only gives a general idea of the search space and does not guarantee that a good result will be found at all. Figure 5.9 shows the total profit change within the first 100 evaluations. The results are completely random and cover a good range.

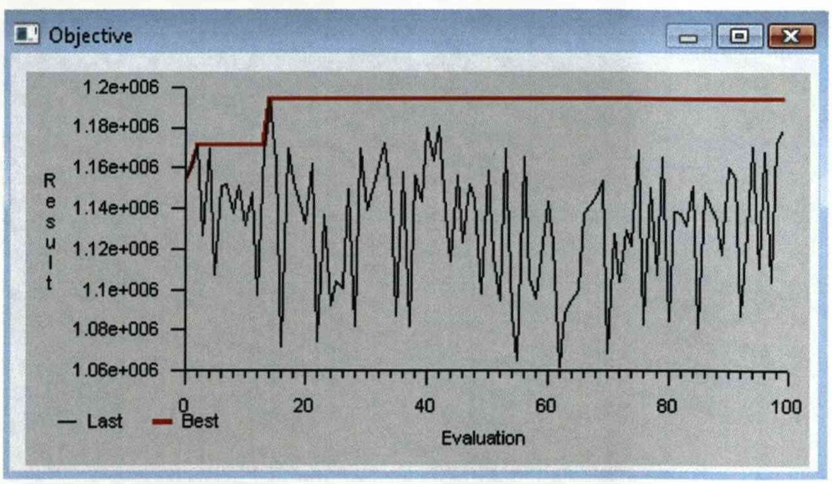


Figure 5.9: The 100 evaluation at Random solutions algorithm.

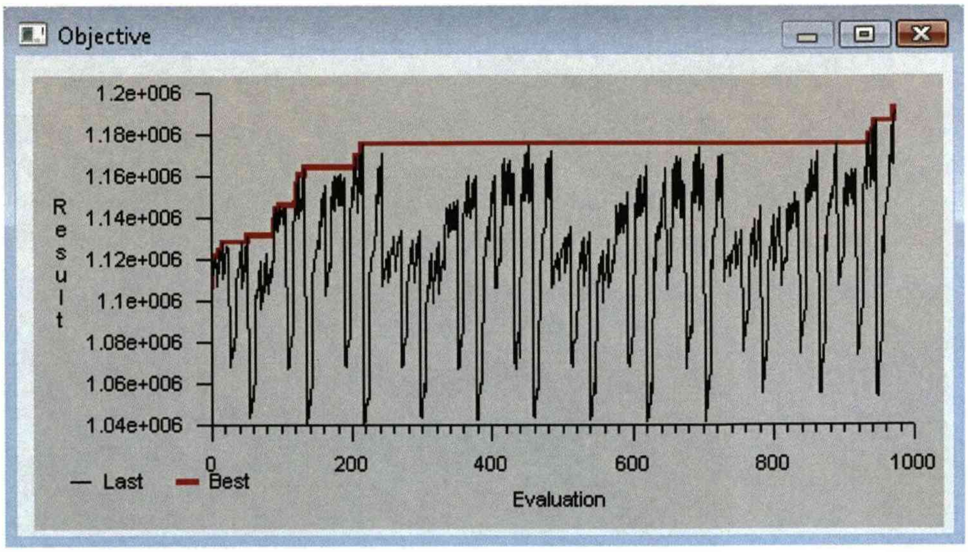


Figure 5.10: First 1000 evaluation of Min/Mid/Max algorithm.

Min/Mid/Max tests the extremes and mid points of all parameter settings. There were five variables per parameter. In this algorithm, only the two end and mid points are tested. Therefore there are three variables per parameter. It brings the total evaluations to 3^7=2178 with this algorithm. It is still not a small number of simulation and it taken nearly two hours for the experiment. The best profit and best throughput results from this experiment do not match the 'all combination' results. This means that the Max points of all parameters do not guarantee the global optima. The benefit and necessity of optimisation experiment is proved again that simply increasing the machine capacity will not always lead to a better system output. Figure 5.10 shows the first 1000 evaluations out of 2178. The figure presents a similar pattern because with the All combination algorithm the two experiments change one parameter from min to max while all other parameters are unchanged. It is a simple method to make sure all combinations are covered throughout the experiments.

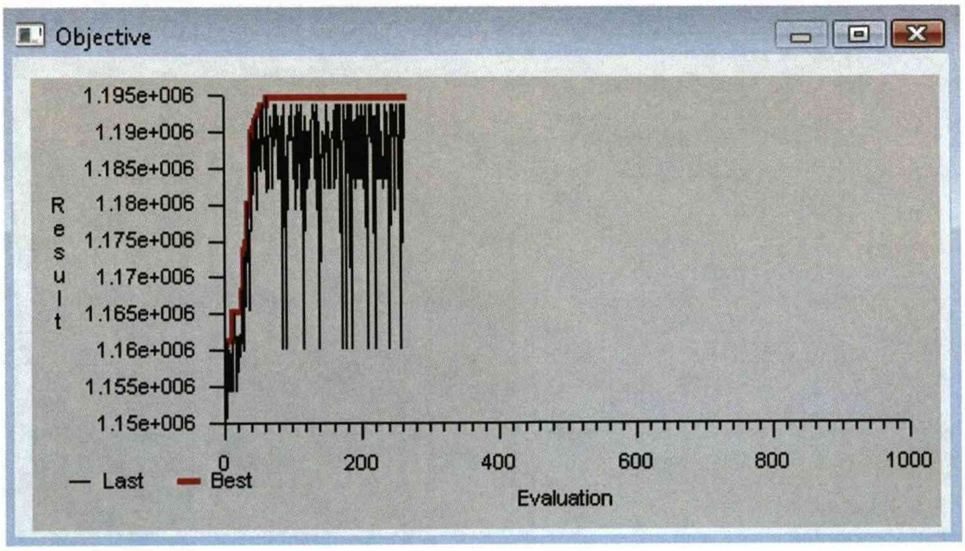


Figure 5.11: Optimisation results using Hill Climb algorithm.

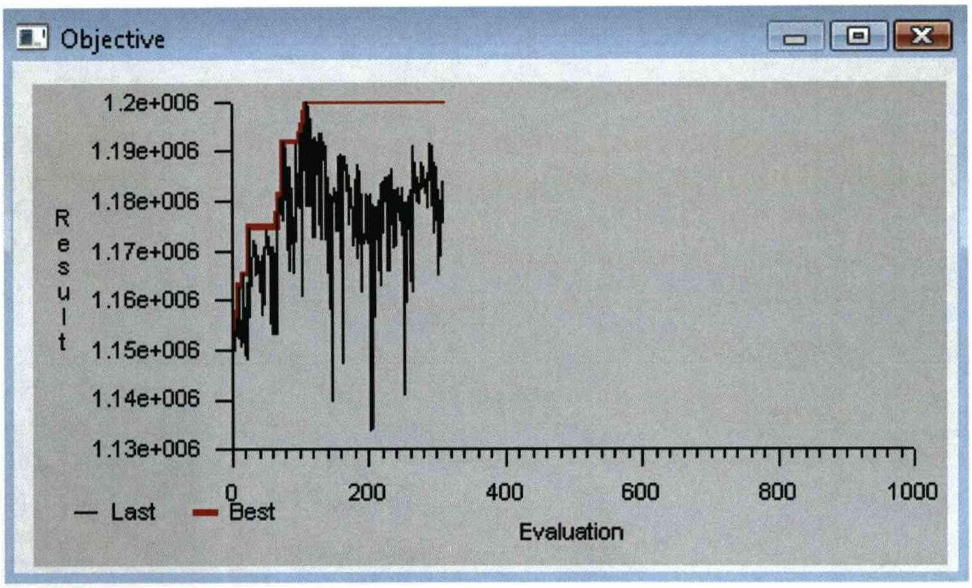


Figure 5.12: Optimisation results using
Adaptive Thermostatistical Simulated Annealing.

Figures 5.11 and 5.12 are the profit optimisation search using Hill Climb and Adaptive Thermostatistical Simulated Annealing. They are both intelligent optimisation algorithms. Hill climb used here is a simple local search algorithm while Adaptive Thermostatistical Simulated Annealing is a global metaheuristic search algorithm. It can be observed from the pattern in Figures 5.11 and 5.12 that these two results do not show the same saw-teeth pattern as the All combinations and Min/Mid/Max do. It is evident that there are intelligent parameter selection methods for these two algorithms. For the above reasons, these two algorithms are chosen for further optimisation experiments.

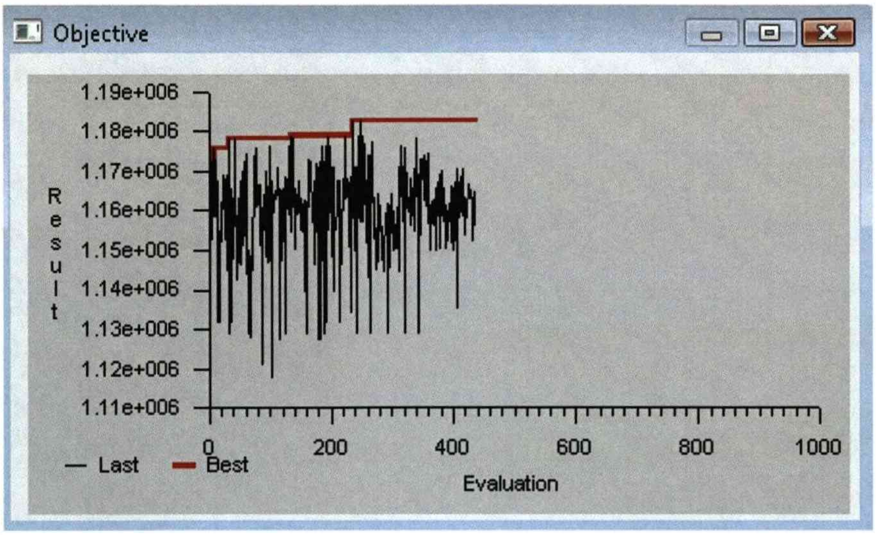
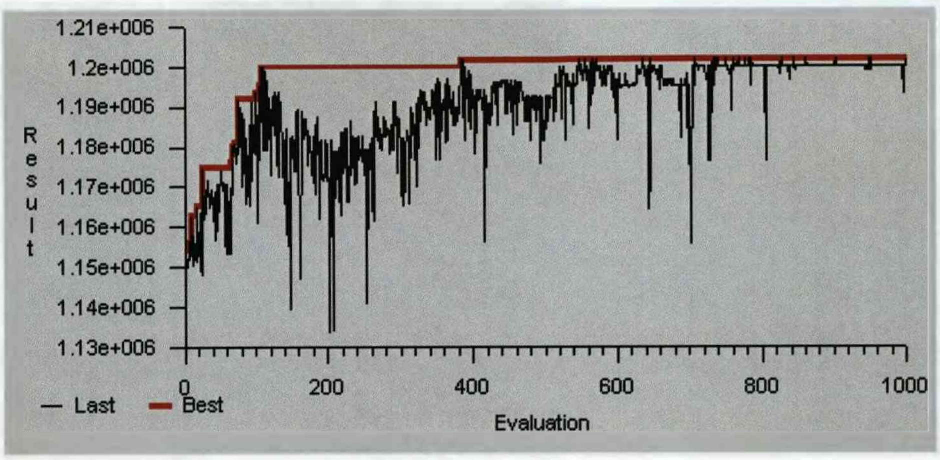


Figure 5.13: Optimisation results using Six sigma.

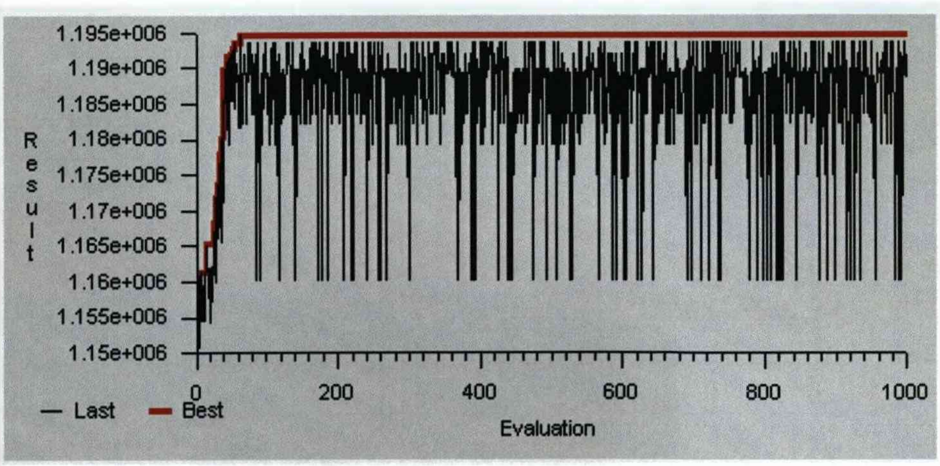
Six sigma algorithm was originally developed by Motorola as a business management strategy and it has been wide-spread in many industrial sectors (J Antony, 2008). Here, the term is used as the name of the algorithm for the reason that the algorithm is developed for the purpose of indentifying the best option for process improvement. The experiment runs 437 evaluations until it reaches the end conditions. In order to run the algorithm a suggested setting is required along with parameter range. The suggested values of parameters were calculated using Theory of Constraints so that all machines meet the capacity requirement. The six sigma search was carried out from the set of suggested values. As the algorithm is designed for process improvement and the scenario here does not fit the profile very well therefore the optimisation results are not as good as other algorithms as expected.

5.6 Further experiments using Simulated annealing and Hill climb

In the previous experiments, the optimisation ended at 265 and 310 evaluations for Hill climb and Simulated Annealing when the algorithms hit the end condition. The default settings for both algorithms are: stop optimisation after 1000 evaluations or 200 consecutive evaluations without improvements. Hill climb found the best profit at evaluation 65 and Simulated Annealing found one at 110 and thereafter no better profit output was found. The max profit found during the optimisation can be identified from the result graph easily. In order to compare the two algorithms and standardise the experiments, 200 consecutive evaluations without improvements was changed to 1000. In this way, all experiments will finish after exactly 1000 evaluations. Each experiment will take about one hour to complete which is the only disadvantage of doing so. Another benefit of changing the setting is that it increases the probability of finding a better result in the extended evaluations. Figure 5.14 shows the result of two optimisation runs after extending the end condition.



(a) Simulated Annealing



(b) Hill climb

Figure 5.14: Optimisation results after extending the end condition.

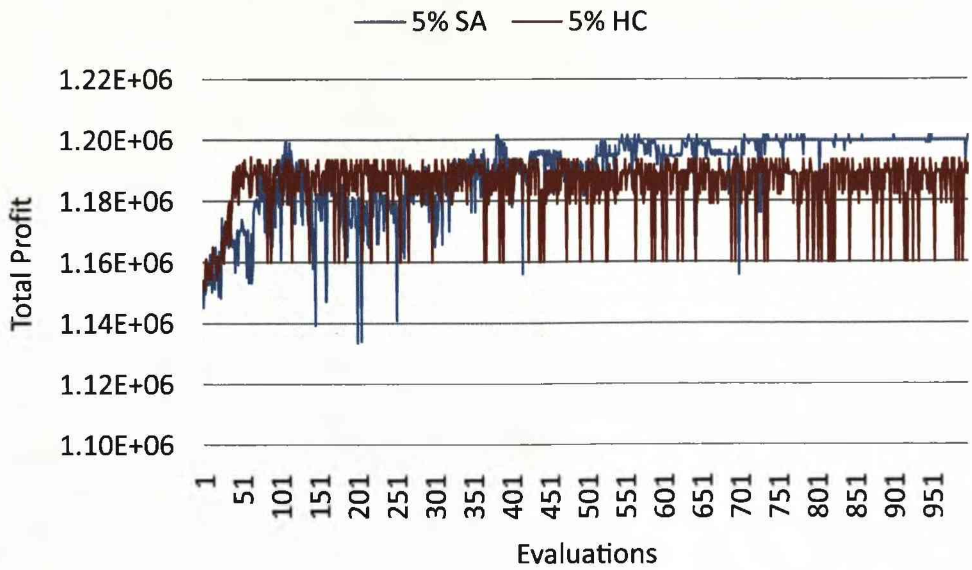


Figure 5.15: Optimisation results of SA and HC at quarter two.

Figure 5.15 combines the results of two algorithms into the same diagram. The blue line is the total profit results of 1000 evaluations using simulated annealing while the red line presents the results by the hill climbing algorithm. Comparing the two algorithms, there are four results that can be concluded:

1. Simulated annealing found a better result than Hill climbing.
2. Hill climbing reached its optima earlier than Simulated annealing.
3. A better result could be achieved when the number of evaluations was increased.
4. As a sign of local optimisation algorithm, the search range of Hill Climbing is limited once the optimum is found while Simulated Annealing shows a bigger range.

The above results are supported by the data in Table 5.5.

Table 5.5: Comparison between Hill Climbing and Simulated Annealing.

Algorithm	Best profit	Best result found at Evaluation	Better result after extending the search	Search range after best result
Hill Climbing	*£1194580*	*No. 64*	*No better result*	*£1160330~ £1194580*
Simulated Annealing	*£1201670*	*No. 383*	*If the search end at No. 310 the best profit is £1199860*	*£1156070~ £1201670*

The conclusions drawn from the experiment results are only based on one set of experiment. It is very arbitrarily to draw conclusions based on one experiment. Further optimisation experiments are necessary and carried out for different periods.

Table 5.6: Parameter settings for optimisation at Quarter three demand.

	MIN	MAX	STEP
percentB	20%	40%	5%
percentC	80%	130%	5%
percentD	0%	20%	5%
percentE	80%	130%	5%
percentF	5%	20%	5%
percentG	20%	40%	5%
percentH	50%	70%	5%
percentI	35%	55%	5%

With the drifting of product life cycle stages, the market demand at Quarter three reaches a peak. Consequently, the machine capacity requirement stepped up by a great deal as shown in Table 4.4. There are eight bottleneck machines in this period. Based on bottleneck analysis, both Machine C and Machine E require 84% and 86% extra capacity. The parameter settings give 80% to 130% improvement for these two machines. In the system reconfiguration, when a machine requires higher capacity, it is sensible to add another machine to achieve a 100% improvement or a better machine to achieve a 100% to 130% improvement. Therefore, the parameter settings proposed are for this practical reason. Table 5.6 gives the parameter settings for the optimisation experiments at Quarter three demand.

Furthermore, Machine D only requires 1.54% of extra capacity from the calculation as shown in section 4.4. It gives a range of 0~20% improvements for Machine D in the parameter settings for optimisation experiment. During the reconfiguration of the manufacturing system, the system output increases when the capacities of bottleneck machines are extended. In order to produce this amount of extra system output, the utilisation of all other machines will go up. The same is applied to bottleneck machines which do not require much extra capacity, for example the capacity requirement of Machine D will increase with improvement of other bottleneck machines. In conclusion it is necessary to set up a higher range for Machine D's improvement plan.

Quarter four has similar complexity as Quarter three; there are 8 bottleneck machines in the system in this period. The parameter settings are shown in Table 5.7. The reasons for such settings are the same as for Quarter three.

Table 5.7: Parameter settings for optimisation at Quarter four demand.

	MIN	MAX	STEP
percentA	0%	20%	5%
percentB	10%	30%	5%
percentC	80%	130%	5%
percentE	80%	130%	5%
percentG	10%	30%	5%
percentH	20%	40%	5%
percentI	50%	70%	5%
percentJ	0%	20%	5%

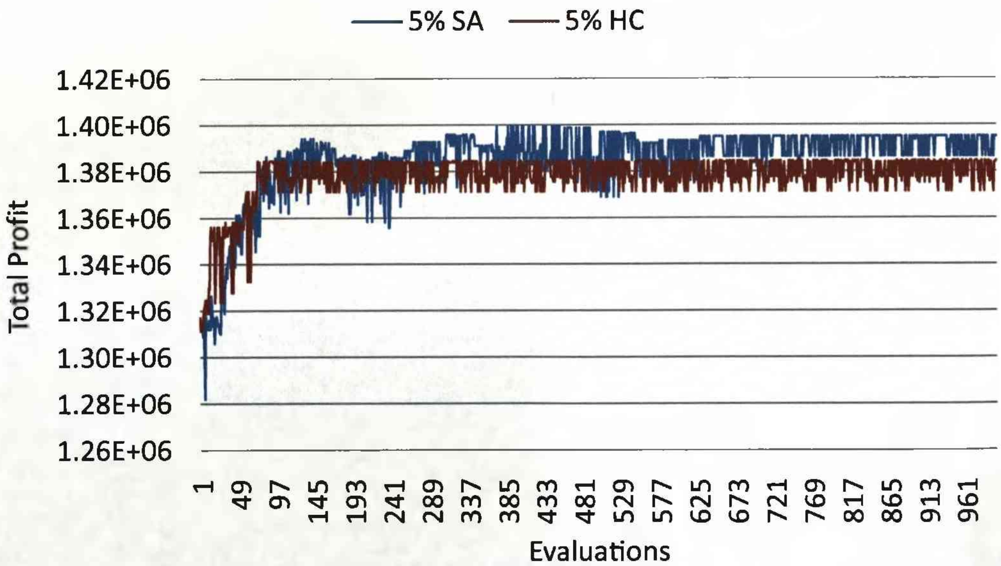


Figure 5.16: Optimisation results of SA and HC at quarter three.

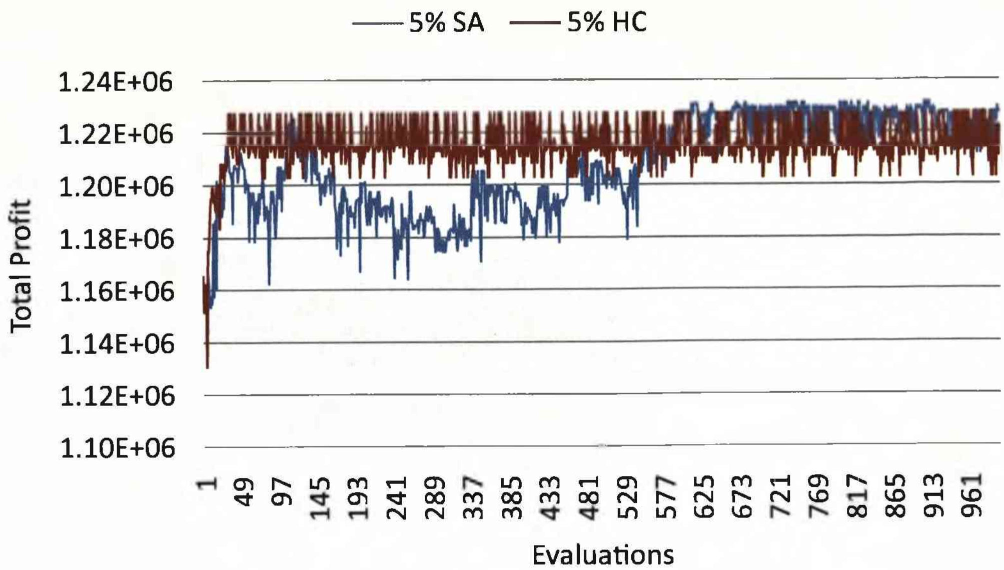


Figure 5.17: Optimisation results of SA and HC at quarter four.

The optimisation results for quarter three and quarter four can be found in Figures 5.16 and 5.17. The top of blue lines are higher than the red lines in both figures indicating that Simulated annealing found better results than Hill climbing algorithm. The red line representing Hill climbing reaches the peak earlier than blue lines. The best results of simulated annealing are found at around the 400th evaluations and 600th evaluations. After the peak is reached, Hill climbing does not seem to be able to search in other areas. In other words, the search is trapped in the local optimum. The four conclusions drawn about the different of simulated annealing and hill climb have been proved again in these two optimisation experiments.

Table 5.8: Parameter settings for optimisation at quarter five demand.

	MIN	MAX	STEP
percentC	20%	40%	5%
percentE	10%	30%	5%
percentI	20%	40%	5%

The parameter settings for Quarter five is shown in Table 5.8 using the same principles. Quarter five is a simpler scenario as there are only three bottleneck machines and they all require no more than 30% additional capacity. Actually, due to the simplicity of quarter five demand request, there is no difference on the performance when choosing different optimisation algorithm. The system will run the 125 evaluations whichever algorithm is chosen for.

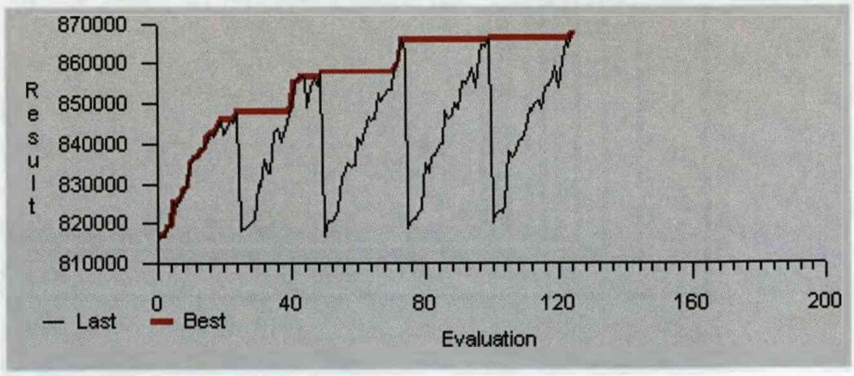


Figure 5.18: Optimisation results at quarter five demand.

Figure 5.18 shows the optimisation results for Quarter five scenario. The optimisation algorithm chosen for the experiments is simulated annealing, however, the results obtained are identical compared with the all combinations or hill climbing method. It is noticed that the profit is still increasing when the higher improvement parameters are used. This is another demonstration showing that TOC suggested capacity change cannot guarantee to meet the market demand. Due to the complexity of manufacturing system process planning, it is unavoidable that machines have idle time. The suggested solution is to combine process planning with reconfiguration optimisation.

5.7 The impact of parameter step size

In the discrete optimisation experiment, parameter settings involve two steps. One is setting the search range for each parameter, the other is the decision of the step size. In the previous experiments the step size is set to be a constant of value 5%. It is commonly understood that when the step size is too small, the simulation will take long time while the large step size diverge the experiment result (Ng et. al., 2001). However, when an algorithm is used for optimisation, the number of total combinations will not have major impact on the computer time any more as the algorithm will still end when the searching conditions are met. Obviously the search results are not the same with different step size. The following simulation experiments aim to find the impact of parameter step size.

5.7.1 Changing step size at quarter one demand

Quarter one represents a simple scenario because there is only one bottleneck machine. Multiple bottleneck machines increase the complexity of the problem exponentially. PrecentF is the only parameter in the optimisation. As demonstrated from the previous experiments, Machine F is no longer a bottleneck when the improvement reaches 25%. Table 5.9 shows that different step size changes the total combinations. Because of the simplicity of this scenario, it only takes less than 2 minutes even when the step size is the smallest.

Table 5.9: Experiment results on different step size.

Step size	2%	5%	8%	10%
total combinations	14	6	5	4
run algorithm	All combination			
total run time	00:01:28	00:00:32	00:00:20	00:00:14

Optimisation results for the above simulation experiments showing all four step sizes are depicted in Figures 5.19 and 5.20 from two different perspectives.

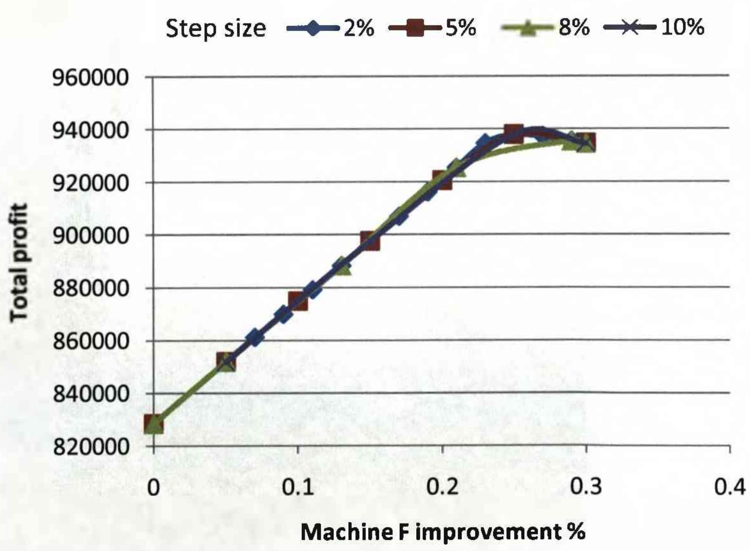


Figure 5.19: Optimisation with different step size on machine F's improvement.

In Figure 5.19, the percentage of machine F's improvement is used as the x axis. The plots of the different step size overlap because of the linear relationship between the profit and the improvement before the peak. However, it can be observed that 8% step size diverged from the optimal point because with 8% step size, the simulation does not simulate 25% improvement on Machine F. In conclusion, the large step size may diverge from the optimal result.

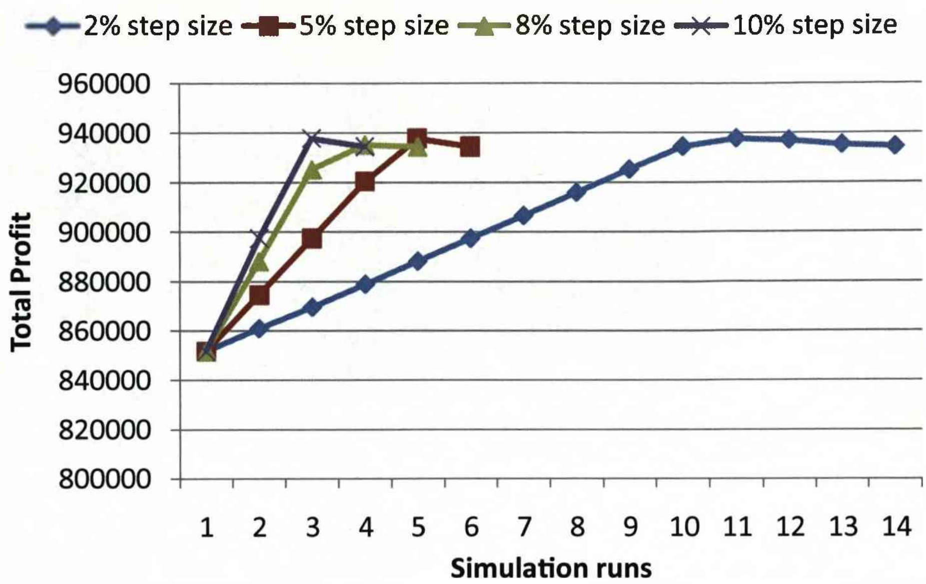


Figure 5.20: Optimisation with different step size, view by evaluations.

Figure 5.20 shows the same experiment result with an evaluation axis. The four result lines in Figure 5.19 are separated in this way. The lower the step size, the more simulation runs are required. The higher the step size, the faster it finds the optimum. However, larger step size will also increase the risk of missing the real optimum.

5.7.2 Changing step size for Quarters two, three and four

From the previous simulation experiments, four conclusions can be drawn for the comparison between the simulated annealing and hill climbing. However, the results are only from one set of experiments. With the change of parameter step size, the results of another 6 groups of experiments are presented in Figures 5.21 to 5.26. The experiments are carried out for quarters two, three and four as they have complicated demand patterns which need an optimisation algorithm to complete the search within the limited computer time. Another two step sizes used are 2% and 8% for the experiments. From these six additional experiments, it can be confirmed that the four conclusions as stated in Section 5.6 are generally correct. Therefore, simulated annealing algorithm has more benefits than hill climbing algorithm.

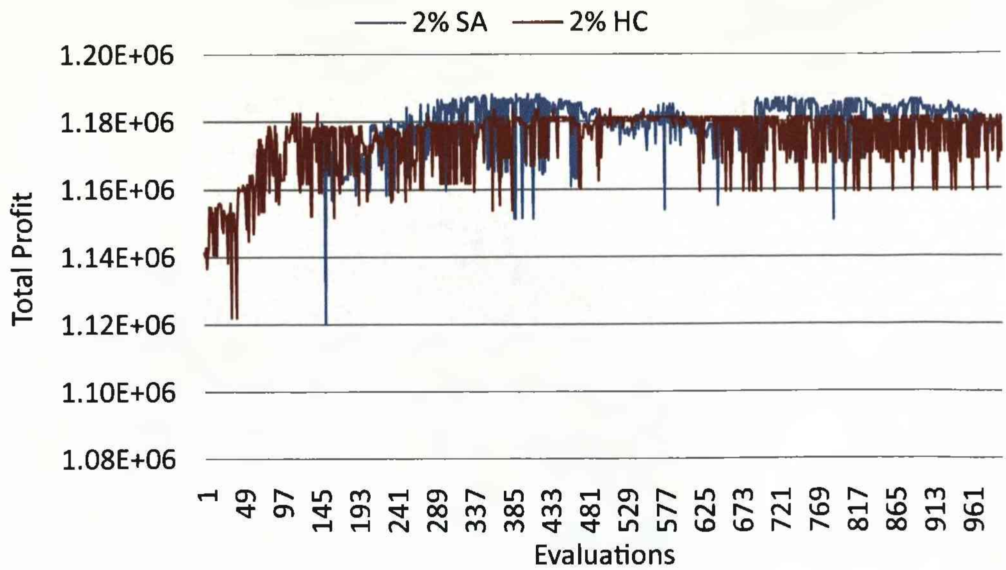


Figure 5.21: Optimisation results at quarter two with 2% step size.

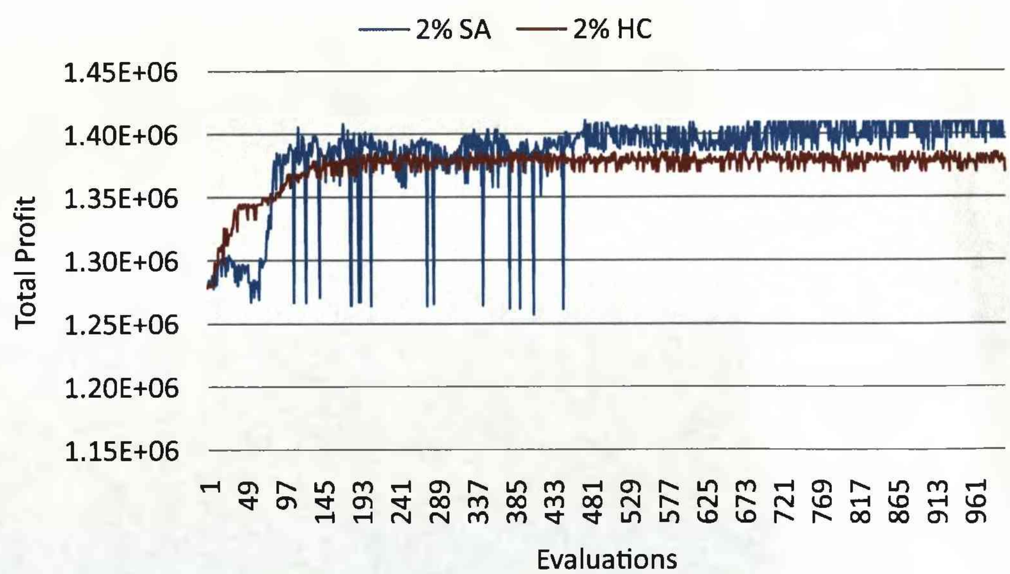


Figure 5.22: Optimisation results at quarter three with 2% step size.

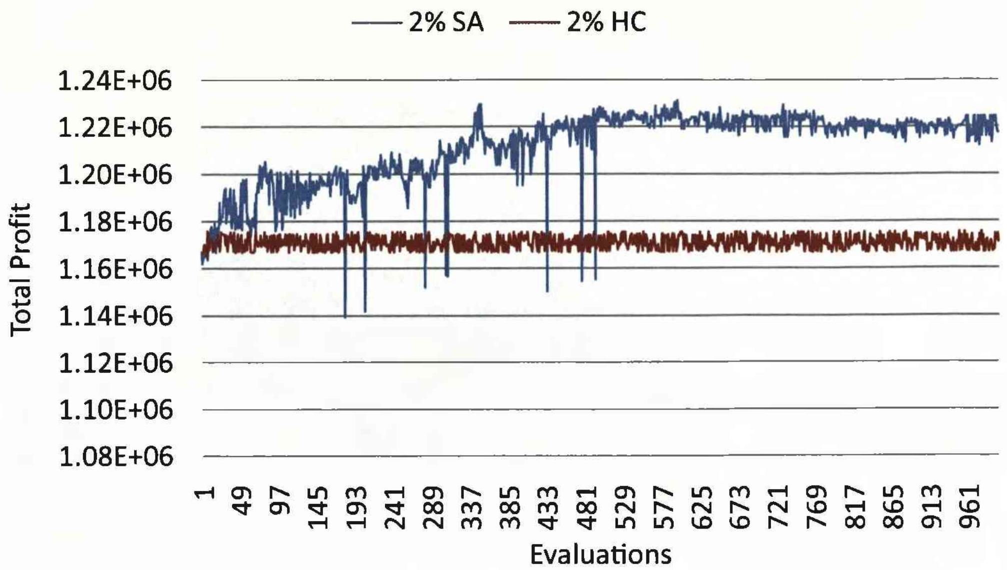


Figure 5.23: Optimisation results at quarter four with 2% step size.

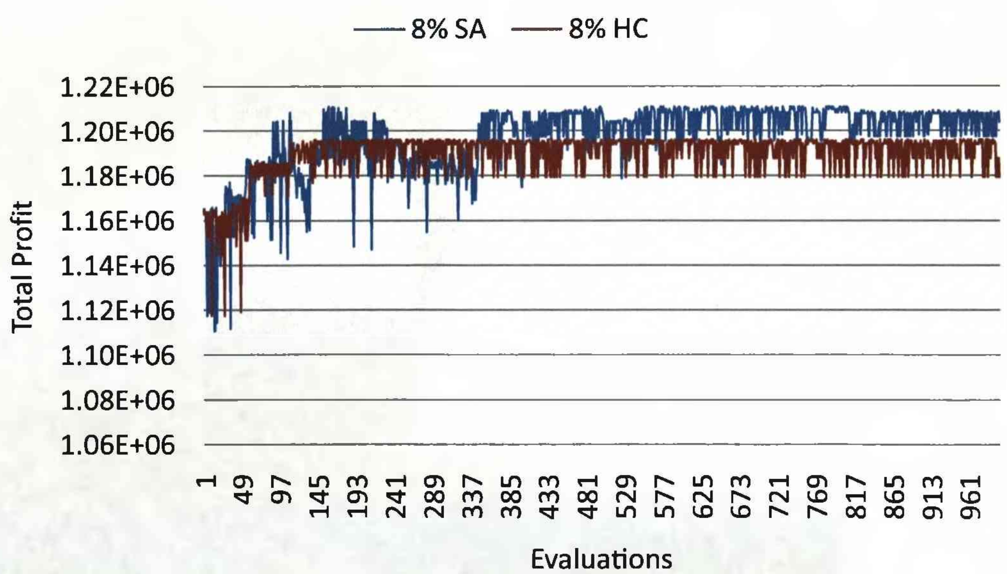


Figure 5.24: Optimisation results at quarter two with 8% step size.

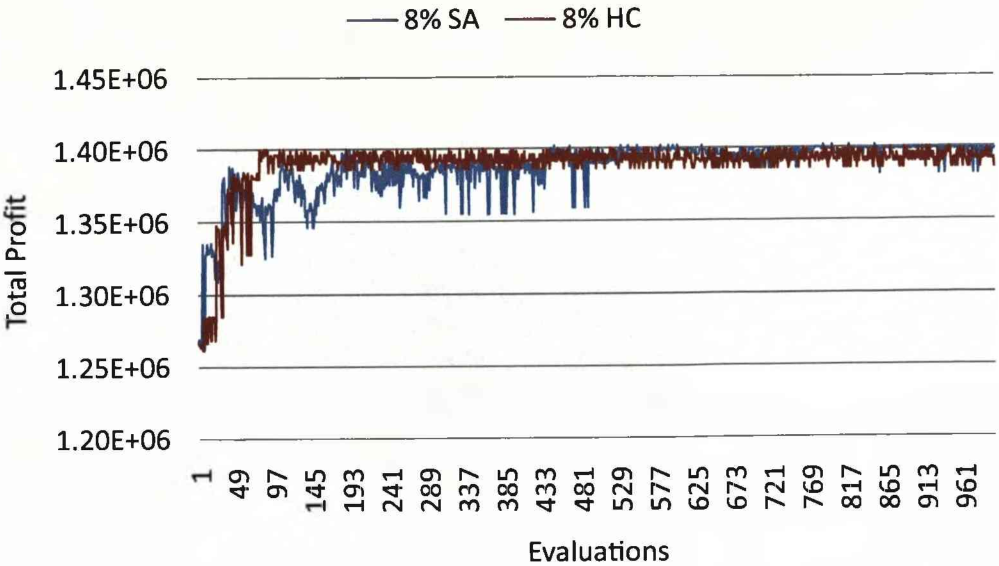


Figure 5.25: Optimisation results at quarter three with 8% step size.

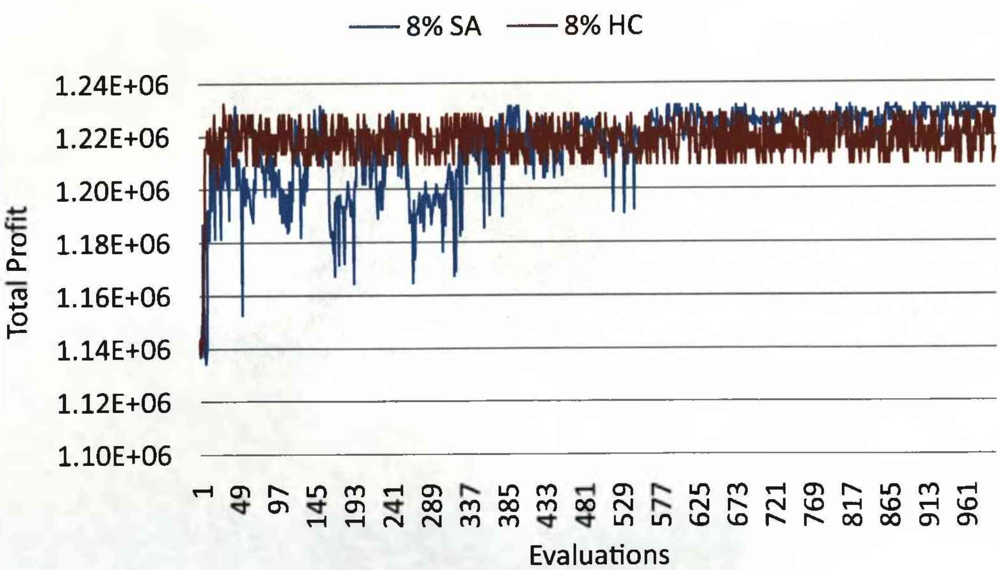


Figure 5.26: Optimisation results at quarter four with 8% step size.

5.7.3 The impact of changing step size

A group of experiments were carried out to identify the impact of changing step size. Step size of 2%, 5% and 8% were chosen to test on three different demand period Quarters two, three and four. Simulation results are given in Figures 5.27 to 5.29 according to the profit obtained at each evaluations. In order to keep the simulation effect constant, all experiments are set to evaluate at least 1000 runs even though the end condition could be met before that. In fact all nine experiments have met the end condition before the 1000th run.

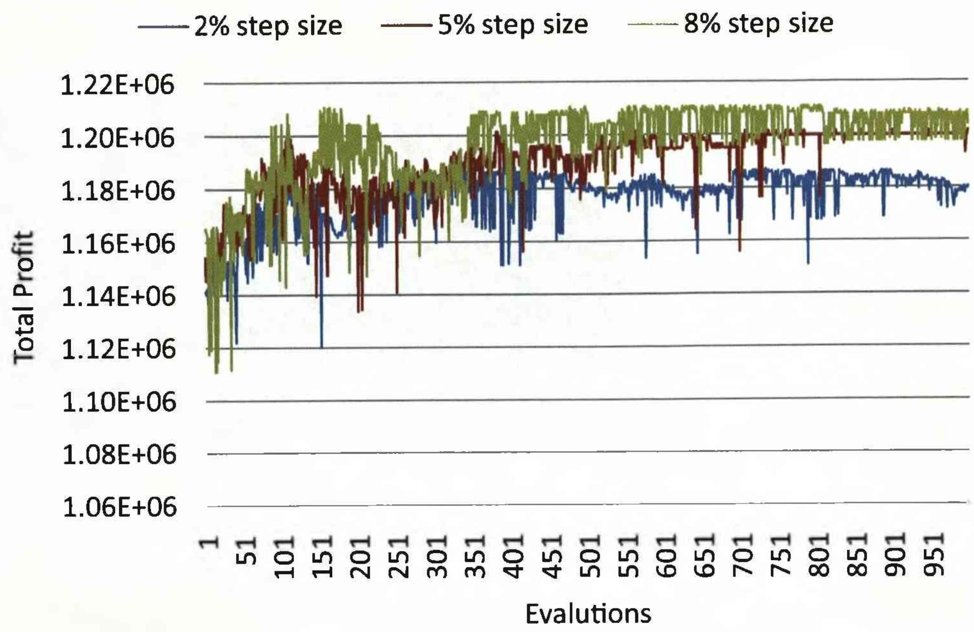


Figure 5.27: Quarter two optimisation trends using simulated annealing.

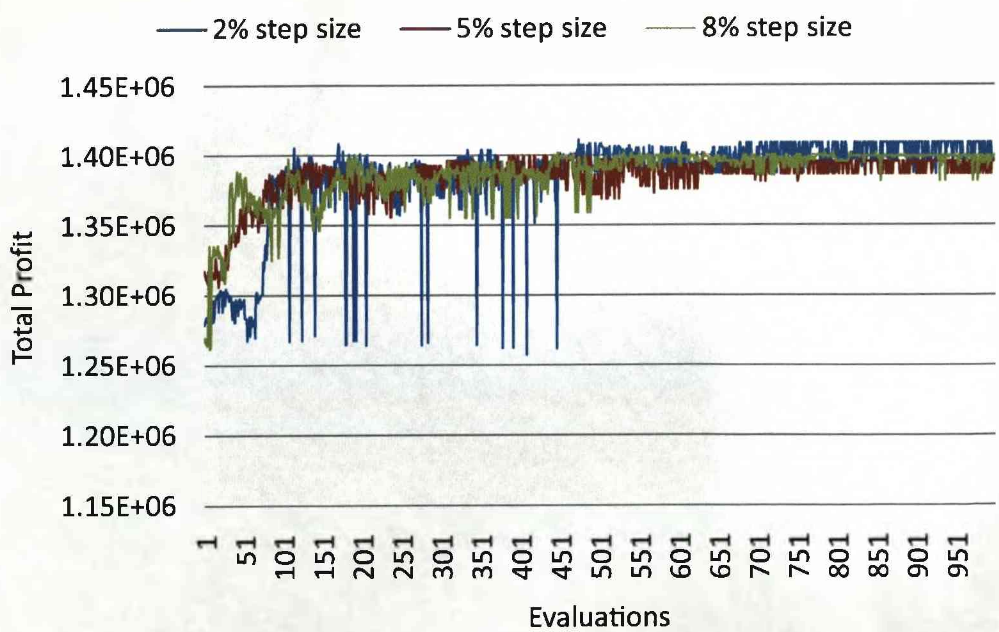


Figure 5.28: Quarter three optimisation trends using simulated annealing.

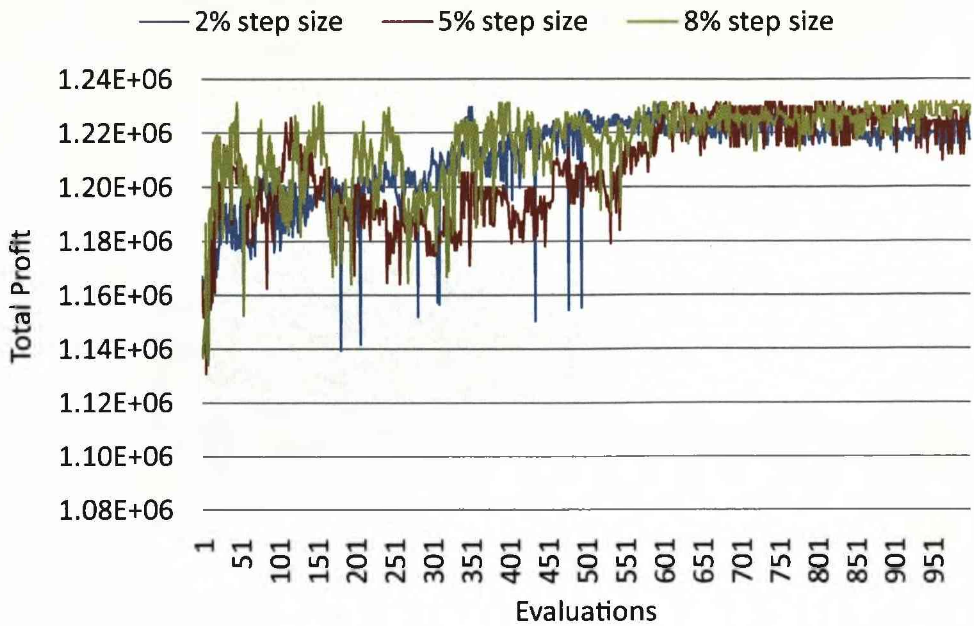


Figure 5.29: Quarter four optimisation trends using simulated annealing.

Simulation results show that changing step size has a direct effect on the optimisation output. The difference it makes is quite enormous in quarter two. It seems that the larger the step size, the better is the result. However, it is only proved for Quarters two and four. The impact of step size in Quarter three is not obvious. In order to find out whether this is a general property of step size, a new step size of 10% is used and the results are added into the previous diagram as shown in Figures 5.30, 5.31 and 5.32.

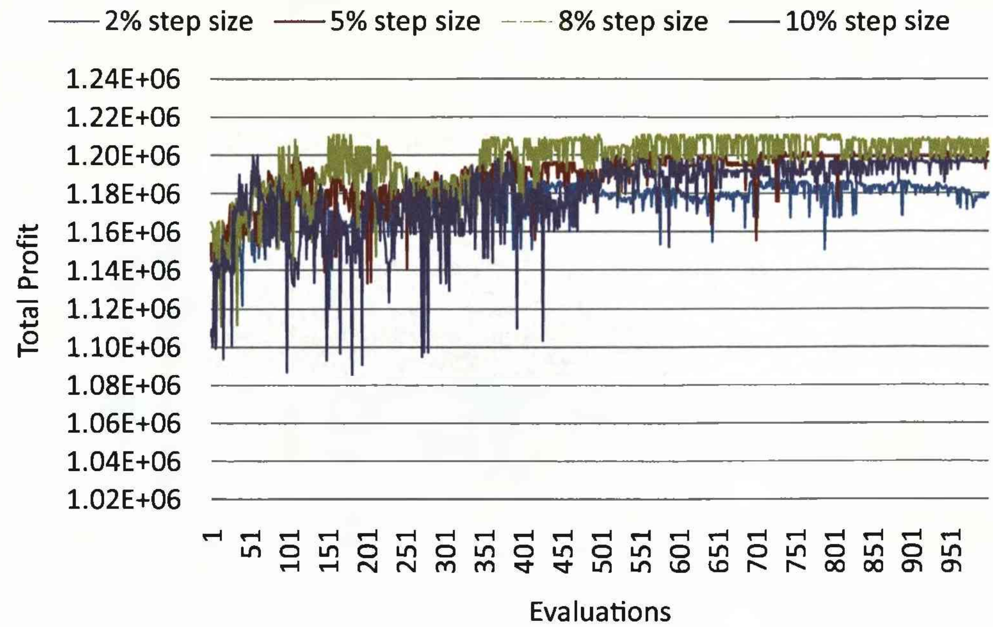


Figure 5.30: Quarter two optimisation trends using simulated annealing.

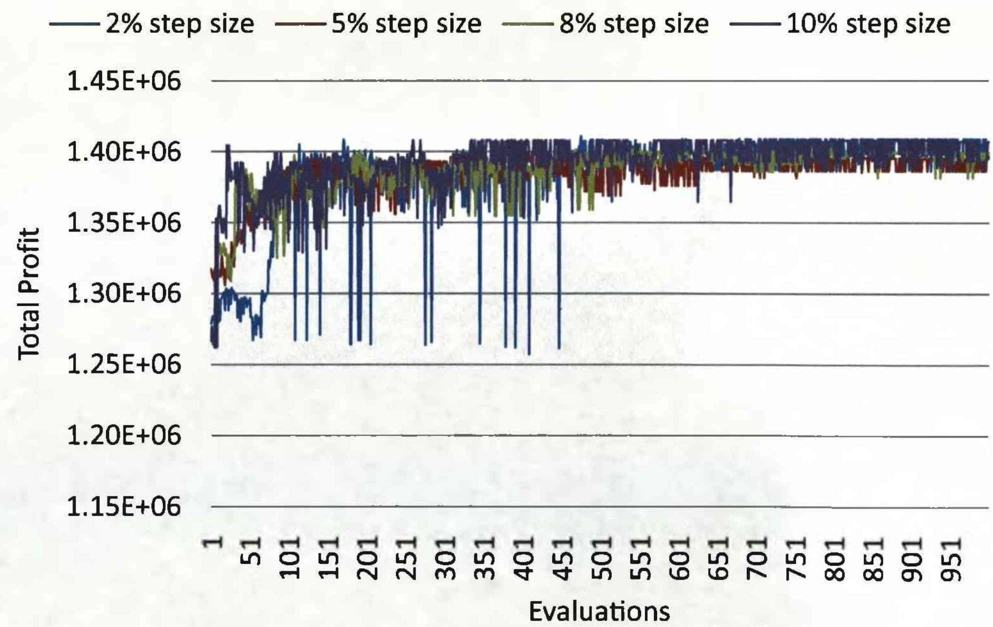


Figure 5.31: Quarter three optimisation trends using simulated annealing.

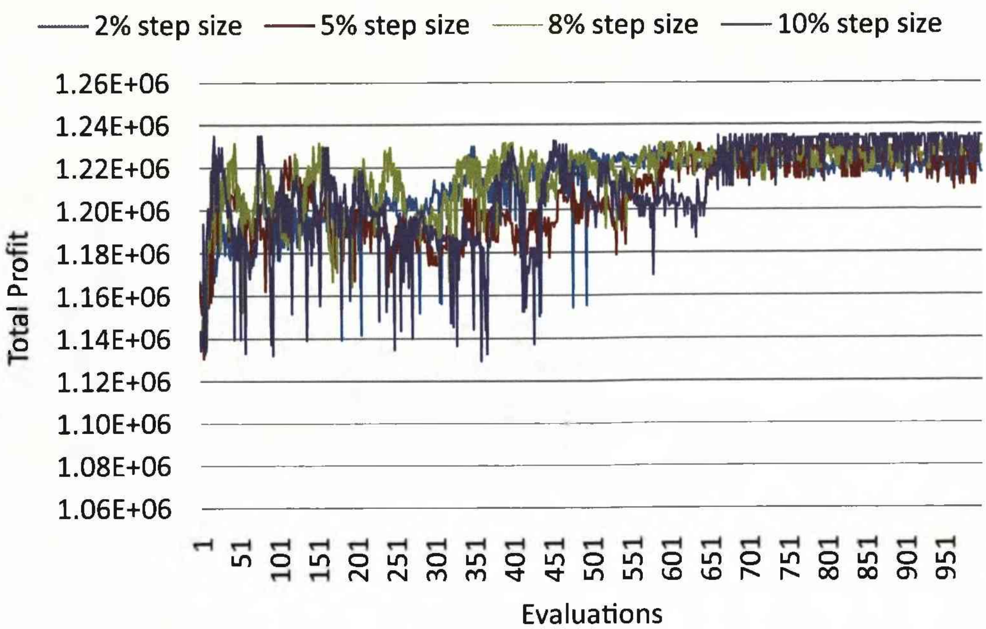


Figure 5.32: Quarter four optimisation trends using simulated annealing.

Table 5.10: Experiment results on changing step size.

		2%	5%	8%	10%
Quarter Two	*Total combination*	19487171	78125	16384	2187
	Best profit	1188090	1201670	1210880	1200360
	Best result obtained at evaluation	No. 361	No. 383	No. 159	No. 56
Quarter Three	*Total combination*	1068910128	2268750	196608	34992
	Best profit	1411090	1399760	1402220	1407940
	Best result obtained at evaluation	No. 475	No. 373	No. 448	No. 260
Quarter Four	*Total combination*	655922124	1210000	110592	26244
	Best profit	1231350	1231400	1231590	1235180
	Best result obtained at evaluation	No. 597	No. 670	No. 47	No. 20

Table 5.10 summarises the experiment data collected from the optimisation test for changing step size. The following conclusions are observed.

(1) Changing step could easily complicate the optimisation problem and enlarge the searching space. For example, from 10% to 2% in Quarter three, the searching space is enlarged by 30547 times.

(2) The smaller the searching space, the shorter time it takes to find the best result. Although it is not strictly true in every case, it is true for most of the cases as the simulation time is also dependent on the algorithm used.

(3) Changing the step size is likely to find different results; nonetheless the value difference of the experiment output is less than 2%. In this particular case, the biggest difference made is in Quarter two between 2% and 8% which is about 1.8% improvement of the best profit. The difference is more subtle in the other two demand periods Quarters three and four which is less than 1%.

In conclusion, it is not very critical to choose step sizes for optimisation search as the difference in results are often very small in optimisation output. Therefore, it is suggested to choose step size according to the available simulation time. For example, if there is time for 1000 evaluations, the step size should be chosen so that the evaluations are no less than 5% of the searching space.

Chapter 6
Cost of the Reconfiguration of Manufacturing system

6.1 Introduction

In manufacturing reconfiguration, there are two key questions: when to reconfigure and how much to reconfigure. In today's global market and competition, the cost of reconfiguration is also a key part of consideration. Balancing the cost of reconfiguration, the decision maker needs a cost model to assess the situation on the system level as well as the machine level.

A cost model which operates upon the previous optimisation results was proposed to assist the decision making. The contents of this chapter are organised as follows: Section 6.2 introduces a reconfiguration cost function which separates the hard configuration cost and the soft cost. Nine linear and non-linear cost models are established in Section 6.3. The last section gives two examples of how to use these cost models to obtain the information for reconfiguration costs.

6.2 Reconfiguration cost function

6.2.1 Background of reconfiguration cost model

The cost function, C(v), introduced in Chapter 2 represents the cost of having a capacity level v. The reconfiguration of a manufacturing system is all about changing the system capacity to meet the required market demand. The cost of changing physical capacity and the other associated non-physical costs such as management and operation cost are always distinguished in the cost function available in the literature. Thus, the cost for each demand period is mainly the cost of having a capacity level v in that period to satisfy the market demand.

ElMaraghy and Shabaka divided manufacturing systems reconfiguration activities into two types: hard and soft. Examples of hard (physical) reconfiguration activities include adding/removing of machines, changing material handling systems. Examples of soft (logical) reconfiguration activities include re-programming of machines, re-planning, re-scheduling, re-routing, and increasing/decreasing of shifts or number of workers (Shabaka and ElMaraghy, 2004). Generally speaking, the system level reconfiguration reflects the cost, time and effort required to perform system level activities that are associated with the reconfiguration process. They are the most expensive activities during the reconfiguration as they mostly involve hard-type reconfiguration, e.g., the adding or removing of machines or stations, installation of materials handling equipment corresponding to a new manufacturing layout. Machine level activities are the next costing elements during the reconfiguration, which involve both hard-type activities, e.g., adding/removing of machine modules or changing machine tools and soft-type reconfigurations activities, e.g., changing of operation scheduling; increasing/decreasing the number of assigned operators. All other activities are called soft-type reconfiguration activities, e.g. buying/selling of machine modules/machines/stations, changing operation process. It is all these activities added together which determine a complete and successful reconfiguration of a manufacturing system.

When considering reconfiguration cost, the different types and levels of soft reconfiguration activities are mostly case-based and cannot be generalised to accommodate all practical situations. It is not only related to the scale of reconfiguration but also a function of the location, industry of the manufacturing system as well as the timing. The infrastructure setup in the facility and the degree of modularity being used on the system and machine level has a great effect on the reconfiguration cost as well. For instance, it is easier to relocate a machine where there was one rather than building a new station for the machine as all the electrical supplies are there already.

The system level and machine level reconfigurations are more deterministic. For example, the reconfiguration can be achieved through adding/removing another spindle to a machine, adding/removing a machine, or even adding/removing a group of machines. All such installation costs can be assessed by the following method. The installation costs of a single capacity increment of size x are assumed to be given by a cost relationship in the form of a power function (Manne, 1961):

$$kx^a \quad (k>0;\ 0<a<1) \qquad [6.1]$$

Manne used the above function to compare the cost of reconfiguration in the pipeline industry at different regeneration point. If, for example, a= ½, this cost function says that a pipeline capable of handing 16 years worth of growth in demand is only twice as expensive as one that can accommodate the demand for four years. The above equation has not been used to calculate the actual machine capacity changing cost, it is used to compare and optimise the time of investment in order to find the best reconfiguration point and the scale the reconfiguration.

6.2.2 Cost function for the reconfiguration of manufacturing system

This reconfiguration model stems from the format of the optimisation results generated via the WITNESS simulation model. The optimisation results determine the scale or degree of reconfiguration required for each individual machine. In addition, this information is in the format of machine improvement percentage. Furthermore, the percentage is a non negative real number and it is allowed to be greater than 1. A reconfiguration cost model is built to convert these numbers to the reconfiguration cost and eventually gives the manager a good idea of how much it will cost to make a serial of changes.

In equation 6.2, C represents the total cost of reconfiguration and the cost function is as follows:

$$C = \sum_{i=1}^{n} k_i x_i^a + \sum_{i=1}^{n} CR_i \qquad [6.2]$$

Where,

i denotes the machine index;

n is the number of machines which require reconfiguration;

k is a cost gain;

x is the percentage of reconfiguration improvement;

a determines the trend of a single machine's reconfiguration cost;

CR represents other soft costs of reconfiguration that is associated with the machine scaling.

The above cost function separates the machine reconfiguration cost with the associated soft costs incurred during machine reconfigurations. Basically, these include other related cost parameters, such as the cost of downtime to rescale the system or to ramp up the new configuration with the new capacity, the operator cost involved and the effort required for that reconfiguration or scaling. The cost of reconfiguration can be a function of the capacity change. The difference between reconfiguration levels can reflect the degree of soft reconfiguration carried out along with the capacity configuration. For example, if the extra capacity required to meet the market demand is minor, an additional spindle or a set of tooling can satisfy the need and complete the reconfiguration, on the other hand, an additional machine or machines may be required if the amount of capacity required is large. In terms of other associated soft cost, the cost of adding a spindle is much lower than adding a machine or station to the system. Therefore, the soft cost can be a linear function multiplies a constant which reflects the proportion of the two costs. This constant is a variable changing with different applications.

Generally speaking, the soft cost of the reconfiguration is a lot smaller than the hard cost and easier to calculate once there is general knowledge on the value of the constant for a particular system. Therefore, this research focuses on the study of machine capacity reconfiguration cost.

6.3 Linear and non-linear reconfiguration cost models

The machine capacity configuration part of cost simply depends on parameter k and a. By varying the parameter a, the cost function can be either linear or non-linear and also can be concave or convex. Several researchers claim that the cost function is concave for their investigated industries (Manne 1961, Luss, 1982). However, they only claim that most of the functions within their research are concave. With the increasing complexity of the manufacturing systems, it is hard to say that all functions are concave and no one has proved that conclusion either. For that reason, in this study, both concave and convex cost functions are studied.

In equation 6.2, the cost gain parameter k determines the slope of the cost function and it varies between deferent machines and the machine cost. In this research, the parameter k is randomly generated as real life data is not available for the research. The advantage of the random generated k is that the research can cover all possible scenarios. The parameter a determines the trend of reconfiguration cost; in other words, it decides the shape of the cost function. If $a < 1$ the shape of the cost function is concave; if a=1 there is a linear relationship between reconfiguration and the cost; if $a > 1$ the cost function is convex.

6.3.1 Using Matlab to generation random parameter *k*

The goal of the cost function is to provide the relationship between the reconfiguration cost and the extra profit generated by adding the extra capacity, simply the relationship between the investment and the return. The results of the cost function will be provided to manager level for supporting their decision-making on system reconfiguration.

The cost gain k is a parameter connected with the actual cost of machine and machine parts. The more expensive is the machine, the higher is the value of parameter k. In reality, parameter k can be determined by the cost of machines and machine parts.

In this research, there is no machine cost information available as the case model was built for theoretical investigation. Therefore, the value of parameter k is randomly generated in the cost function. In many simulation models there is a need to explore the variable to assist the decision-making with certain likelihoods. Random number generation is included in WITNESS to enable this function. However, computer languages can only use an algorithm to generate random numbers rather than true random number generated by rolling a dice or tossing a coin. WITNESS generates pseudo random numbers by using a combined multiple recursive generator. This method generates random numbers between 0.0 and 1.0. In order to calculate a different number each time, the pseudo random number generator uses the previous result to form the basis for calculating the next. In this way, streams of random numbers are formed and replications of random number can form the distribution of the total likelihoods. However, the method always uses the same start number at the beginning of the simulation runs. In our case, all parameters k for each machine are generated before the start of the simulation and they will be the same for each simulation run. This is not suitable for the purpose of the research as a random k is required at each situation.

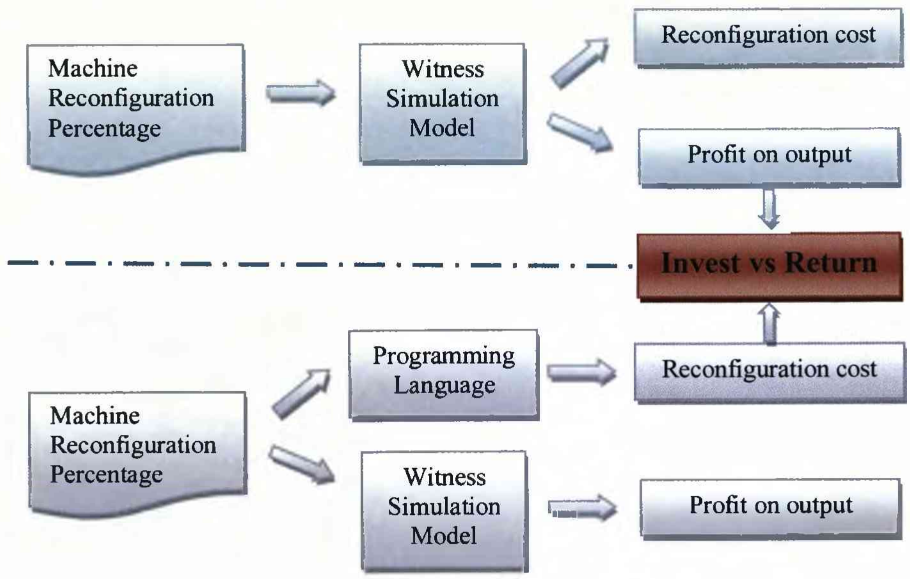


Figure 6.1: Use of WITNESS and Matlab in the investigation.

For the above reason, Matlab was used to generate random number for this particular part of research. Figure 6.1 demonstrates the purpose of using Matlab. The system output and the profit on the output is generated by WITNESS model which does not change with the reconfiguration cost function as the profit on output is a function of sales and material costs. The cost function calculates the reconfiguration cost using the machine reconfiguration percentage which is also the input information of WITNESS model. Therefore, the cost function can totally run outside of WITNESS model and generate the information of the reconfiguration cost. The decision maker can then gather the information on reconfiguration cost and profit on new output and to compare the investment and return. The two methods of getting random parameter k result the same decision making information. The complete Matlab code for the cost model is attached in appendix D.

6.3.2 Linear and non-linear cost model

The cost of a typical machine capacity change can be simply calculated by kx^a from equation 6.2. This does not consider other costs associated with the capacity change. The parameter a in the above calculation determines the shape or trend of the cost model. In this research, nine possible situations are studied for the cost models which include three linear scenarios and six non-linear scenarios.

The value of parameter a was set to 1, ½ and 2 for the linear, non-linear convex and non-linear concave scenarios. In real life situation, the value of a, or even the cost model itself is rather uncertain. The above three values can represent and demonstrate the difference between linear, non-linear and convex, concave scenarios. Therefore, the following experiments are carried out based on the above a value. The uncertainty in each real life reconfiguration case are affected and amplified by many other reasons. The following analysis gives an example of the most likely scenario and provides the background knowledge when dealing with a specific situation.

In the current UK machine tool market, a low cost machine is around £20,000 to £30,000 and a high cost machine is about £50,000 to £75,000. Certainly, machine costs have a bigger range than the above figure. The cost model price range can be adjusted by the settings of value k according to actual costs. In this research, the machine costs are set to the above range for low or high cost machines. The non-linear convex machine cost model is chosen as an example which is displayed in Figure 6.2.

Figure 6.2 shows the plot of function kx^a, where k is randomly generated by Matlab from range 2 to 7.5 to uniform machine cost from £20,000 to £75,000 and a is set to be ½ to represent the non-linear convex cost model. The x-axis states the improvement of the machines, i.e. the value of the machine capacity percentage change. The range of x-axis is set to be 0% to 100%. In the case of higher improvement percentage, the range can be extended in a reasonable degree. The cost of reconfiguration does not simply follow the function all the way to a higher percentage. For example, a 200% improvement means a purchase of another two machines. At the 100% improvement the cost of reconfiguration is the cost of buying another machine. It is a lot less than the cost of two machines if the cost function close to 200% is simply calculated by the extension of the above cost function. Therefore, when the configuration level is higher than 100%, the reconfiguration cost calculation needs to be treated as two reconfigurations rather than reading the cost from one diagram.

The y-axis is the result of the configuration cost for each machine. It is standardised from 0 to 100 for all cost model scenarios and each unit stands for £1,000. There are twelve lines each indicating the reconfiguration cost for the twelve machines respectively. These twelve lines are generated randomly between $2x^{1/2}$ and $7.5x^{1/2}$ so that the cost to purchase a new machine matches the market price of £20,000 to £75,000. The simulation model calculates the machine improvement data and outputs to an Excel spreadsheet. The cost function can use these improvement percentages as the value of x and find the reconfiguration cost for each machine using the cost function. The total reconfiguration cost for the whole system is the sum of each machine's reconfiguration cost.

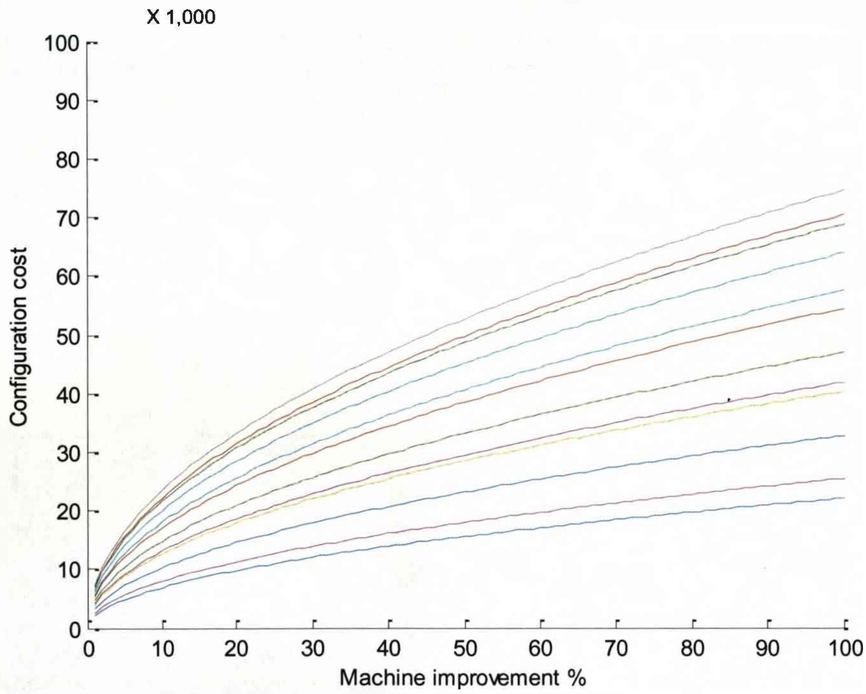


Figure 6.2: Random non-linear convex variable machine cost model.

Figure 6.2 is only one of the cost models proposed in this research. The nine reconfiguration cost models are given in Table 6.1. In the use of the cost model, actual data can be analysed to find out which cost model is the most appropriate for the situation. Furthermore, with accurate machine cost, the value of parameter k can be generated in a more narrow and accurate range.

Table 6.1: A summary of reconfiguration cost models.

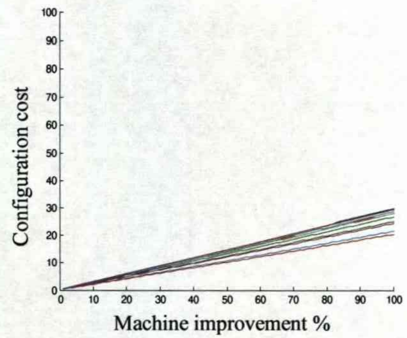
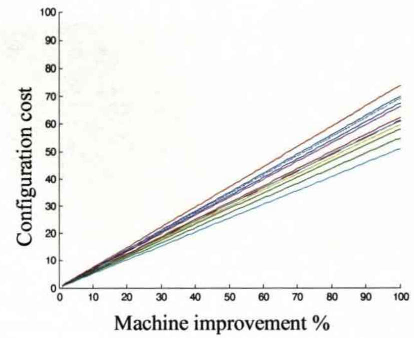
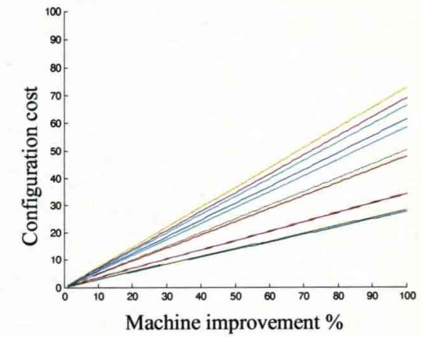
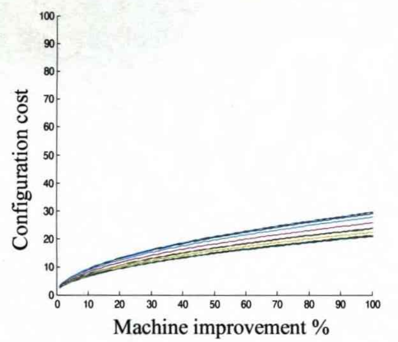
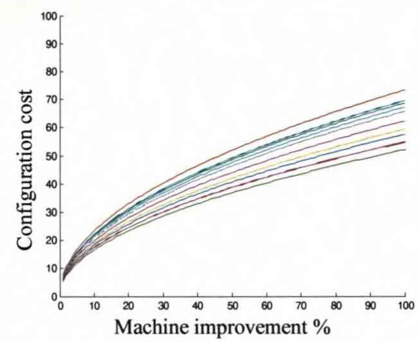
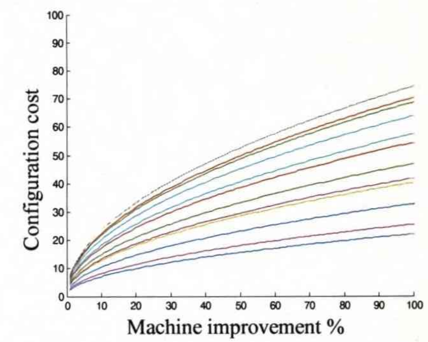
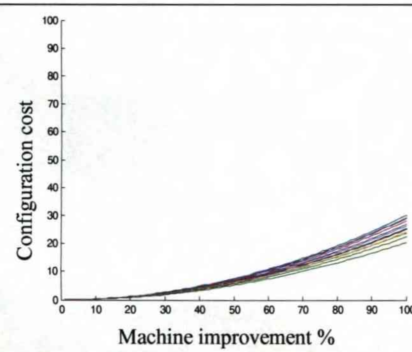
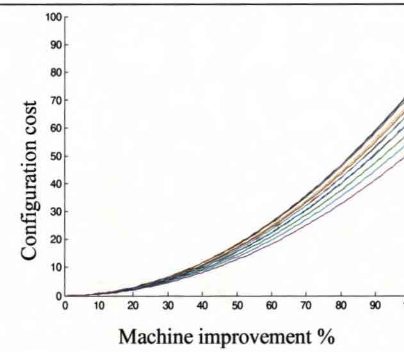
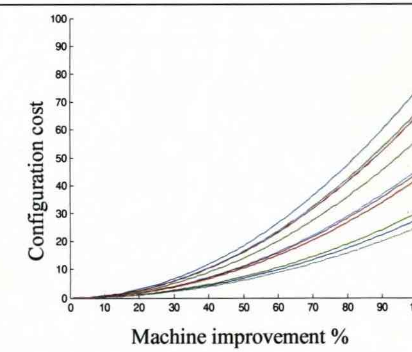
<table>
<tr><th></th><th>Low cost machines</th><th>High cost machines</th><th>Vary cost machines</th></tr>
<tr><td>Linear
(a=1)</td>
<td></td>
<td></td>
<td></td></tr>
<tr><td>Non-linear
convex
(a=1/2)</td>
<td></td>
<td></td>
<td></td></tr>
<tr><td>Non-linear
concave
(a=2)</td>
<td></td>
<td></td>
<td></td></tr>
</table>

Table 6.1 shows a summary of the nine machine reconfiguration cost models. In these cost models, all twelve machines are grouped together under one cost model. It is more realistic for each machine to follow a different cost model and therefore in equation 6.2 parameters k and a are both given a machine number index. Thus, the sum of the total configuration cost would use between one and twelve cost models. A more readable version of the cost models and more examples of cost models are listed in Appendix C.

6.4 Simulation results of reconfiguration cost

The purpose of calculate reconfiguration cost is to give the decision maker the necessary information for system reconfiguration so that a better decision on when to reconfigure and how much to reconfigure can be made. The cost function and the nine proposed models are an efficient method of dealing with the reconfiguration cost. In this theoretical study, the associated soft cost of reconfiguration is not considered. However, it is not difficult to add them into the equation once there is enough information.

6.4.1 An example of analysing reconfiguration cost

The following section will give an example of how to achieve the reconfiguration cost information using the Witness Optimizer and the cost function. Quarter one demand is used in this example as there is only one bottleneck machine requiring reconfiguration which makes it easier to demonstrate the method. The more complicated situation is simply the accumulation of the one machine cost.

In quarter one, machine F is the only bottleneck machine requires reconfiguration. The WITNESS Optimizer gives following results on the reconfiguration of machine F and the consequent improvement of the total system profit.

Table 6.2: Total profit produced with reconfiguration for Machine F.

Machine improvement percentage												Profit	Throughput
A	B	C	D	E	F	G	H	I	J	L	L		
0	0	0	0	0	0	0	0	0	0	0	0	828405	1853
0	0	0	0	0	0.05	0	0	0	0	0	0	851840	1896
0	0	0	0	0	0.1	0	0	0	0	0	0	874730	1938
0	0	0	0	0	0.15	0	0	0	0	0	0	897620	1980
0	0	0	0	0	0.2	0	0	0	0	0	0	920510	2022
0	0	0	0	0	0.25	0	0	0	0	0	0	937950	2054
0	0	0	0	0	0.3	0	0	0	0	0	0	934650	2034

Various cost models propose methods of calculating the reconfiguration cost for machine capacity improvement. Table 6.3 demonstrates the cost of machine F's reconfiguration under different cost models. As there is only one machine requiring reconfiguration, the three cost models which vary the machine costs are not applicable in this case. The table lists all cost figures calculated by the six cost models and the profit return associated with the reconfiguration. Each cost data is the average of 10 random numbers generated by Matlab.

Table 6.3: Reconfiguration cost and extra profit for different cost models.

Machine F's improvement %	**0**	**0.05**	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**
Linear low cost model (LLC)	0	1293.5	2587	3880.6	5174.1	6467.6	7761.1
Linear high cost model (LHC)	0	3412.9	6825.8	10238.7	13651.7	17064.6	20477.5
Non-linear convex low cost model (NXL)	0	4708.3	6658.6	8155.1	9416.7	10528.1	11533
Non-linear convex high cost model (NXH)	0	14076.3	19906.9	24380.9	28152.7	31475.7	34479.8
Non-linear concave low cost model (NVL)	0	65.4	261.7	588.8	1046.8	1635.7	2355.4
Non-linear concave high cost model (NVH)	0	162.1	648.6	1459.3	2594.4	4053.7	5837.3
Return £	0	23435	46325	69215	92105	109545	106245

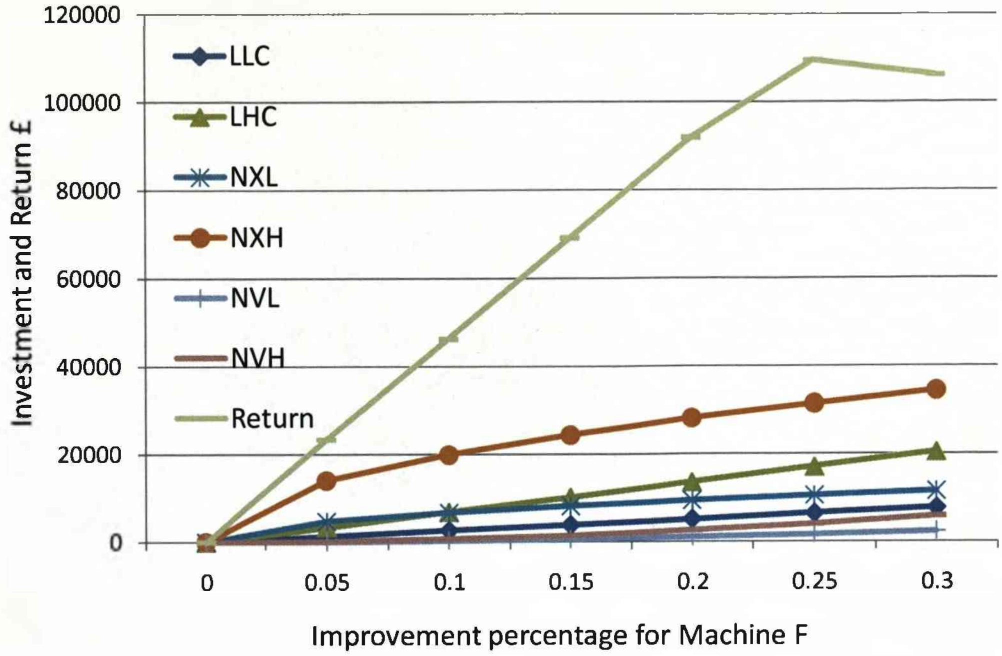


Figure 6.3: Investment against return using six different cost model.

Figure 6.3 illustrates the relationship between investment for machine reconfiguration and the profit return. It is clear that different cost models result in a different configuration cost and with the increase of capacity improvement the cost goes up simultaneously. The non-linear convex high cost model leads to the most expensive machine improvement cost and the non-linear concave low cost model is the cheapest. It is advisable to develop a real life case to decide on the selection of cost model and the percentage of the improvement. However, it is very clear from Figure 6.3 that the profit return goes up by a large amount compared to the machine improvement cost. It is easy to make a reconfiguration decision when the investment and return has such a big difference although the above figure does not take the associated soft reconfiguration cost in consideration. Other facts would also have to be analysed and taken into consideration for more complicated situations.

6.4.2 More complicated reconfiguration cost scenarios

The cost function is a very straight forward mathematical model and the total cost is the sum of each individual machine or station reconfiguration costs. Therefore, the complexity of getting a reconfiguration cost for a complicated manufacturing system is reduced by a large extent. Analysis of each machine/station separately; adding each individual cost together to get the total reconfiguration cost and comparing the cost with the system output improvement are the three main steps on reconfiguration.

The following example illustrates the proposed method for manufacturing reconfiguration cost for quarter two which is a very complicated configuration period with seven bottleneck machines.

Similar to quarter one situation, WITNESS Optimizer gave an optimisation result for the best reconfiguration improvement combinations obtained using a simulated annealing algorithm. Table 6.4 shows five examples of the optimisation results which are among the best 50 optimisation results.

Table 6.4: Total profit produced with machine reconfigurations.

Machine improvement percentage												Profit	Throughput
A	B	C	D	E	F	G	H	I	J	K	L		
0	0	0	0	0	0	0	0	0	0	0	0	907595	2087
0	0.25	0.35	0	0.5	0.35	0.25	0.3	0.4	0	0	0	1196670	2905
0	0.3	0.35	0	0.5	0.35	0.3	0.3	0.4	0	0	0	1198265	2897
0	0.3	0.35	0	0.5	0.35	0.25	0.3	0.4	0	0	0	1199015	2916
0	0.3	0.35	0	0.5	0.35	0.25	0.35	0.4	0	0	0	1199850	2905
0	0.3	0.35	0	0.5	0.35	0.3	0.35	0.4	0	0	0	1199860	2889
					...							...	...

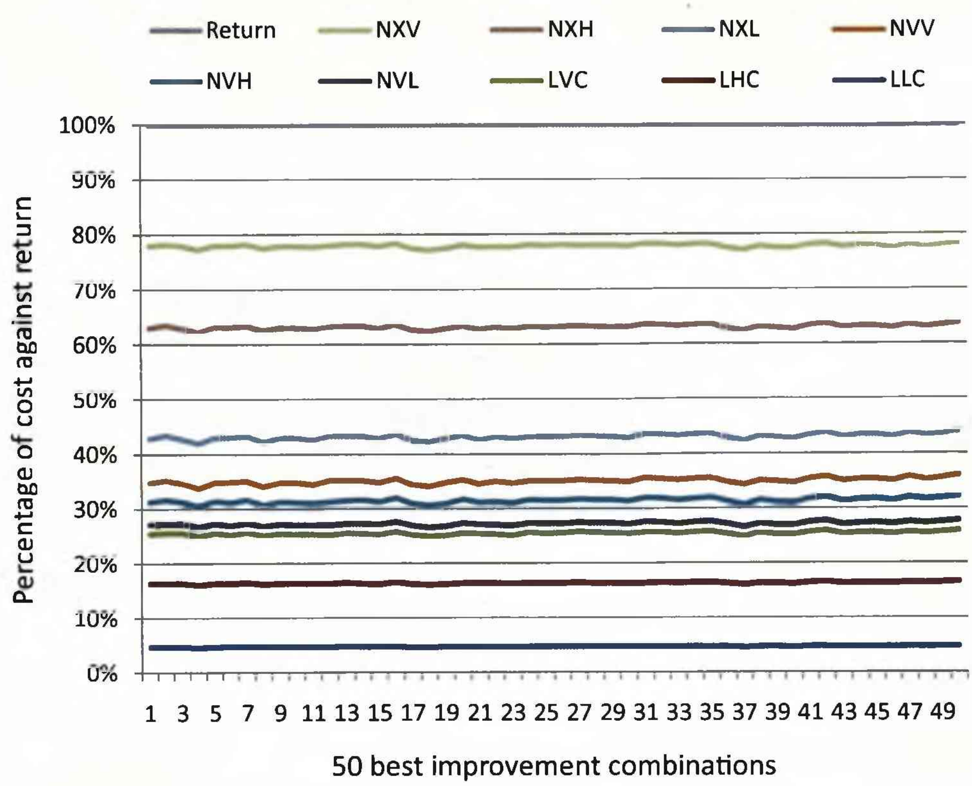


Figure 6.4: Cost of the 50 best improvement combinations.

The 50 best improvement combinations of optimisation results are used as the input data for the cost functions. Using the nine different cost models, the total reconfiguration cost for the manufacturing system can be calculated using the sum of individual improvement costs. Figure 6.4 gives the relationship between the total reconfiguration costs for all nine cost models and the profit return. It is assumed that all bottleneck machines use the same cost model although it is more realistic that different machines follow different cost model. All cost data used in Figure 6.4 are the average of 10 randomly generated cost models.

The accumulation of reconfiguration cost amplifies the effect of the different cost models. Unlike the one machine situations, the total cost of seven machines' improvement can be as little as 5% of the profit or as much as nearly 80% of the profit when choosing different cost models. Uncertainty, in this case, has a stimulating effect upon the magnitude of individual investments.

Chapter 7
Product Portfolio Restructuring for the System Reconfiguration

7.1 Introduction

In this investigation, the simulation methodology consists of three main steps, i.e., simulation of the original configuration, optimisation of the reconfigured system and finally product portfolio restructuring. In the case where the full demand could not be satisfied by the highest permissible level of reconfiguration, the product portfolio optimisation routine will provide priority options to outsource products. Such considerations are based on the utilisation of the existing system, the volume of the removed products and ultimately, the economic impact of the new arrangement. Two product portfolio restructuring strategies on dedicate production line and part outsourcing are discussed in this chapter.

7.2 Product portfolio strategy

The product portfolio is a collection of products under one brand or product line. The best portfolio is one that satisfies the manufacturing system's capacity and maximizes the total profit.

For any manufacturing companies, product portfolio planning constitutes one of the most important decisions regarding how to offer the '*right*' products to the target market and the '*right*' product for production by the manufacturing system. Essentially, such decisions exhibit a typical combinatorial optimisation problem, which deems to be very complex and difficult to solve because of unknown factors in the market and manufacturing responses.

The reconfiguration of manufacturing system has enabled manufacturers to provide a 'cost effective' product range to the market. The cost of manufacturing makes it prohibitive to supply all the possible variants to the market as demonstrated in Chapter 6. As discussed in Section 4.6, even with a very high level of reconfiguration, the market demand is still not met. Therefore, the determination of the right number of product variants to offer in the product portfolios becomes an important consideration. The product portfolio planning problem had been independently studied from marketing and engineering perspectives (Bryan, 2007). However, in different industries, or even in the same industry, the solution for each individual case is different.

7.3 An example of product portfolio reconfiguration

An example of how to make product portfolio decision is presented in this section using the same manufacturing model. The objective of the product portfolio planning is to minimize reconfiguration investment and to optimise the utilisation of the manufacturing system.

Table 7.1: Machine capacity used for product P1 to P5.

Product P1	Q1	Q2	Q3	Q4	Q5
A	105	245	490	861	735
B	90	210	420	738	630
C	180	420	840	1476	1260
D	45	105	210	369	315
E	150	350	700	1230	1050
F					
G*2	120	280	560	984	840
H*2					
I	270	630	1260	2214	1890
J	150	350	700	1230	1050
K					
L					
Sum	1110	2590	5180	9102	7770
Profit	2475	5775	11550	20295	17325

(a) Profit for Product P1 in five quarters

Product P2	Q1	Q2	Q3	Q4	Q5
A	420	756	1380	1296	780
B	210	378	690	648	390
C	665	1197	2185	2052	1235
D	350	630	1150	1080	650
E	735	1323	2415	2268	1365
F					
G*2	630	1134	2070	1944	1170
H*2	1225	2205	4025	3780	2275
I					
J	420	756	1380	1296	780
K					
L					
Sum	4655	8379	15295	14364	8645
Profit	12425	22365	40825	38340	23075

(b) Profit for Product P2 in five quarters

Product P3	Q1	Q2	Q3	Q4	Q5
A	120	234	198	156	105
B	800	1560	1320	1040	700
C	520	1014	858	676	455
D	440	858	726	572	385
E	440	858	726	572	385
F	480	936	792	624	420
G*2	960	1872	1584	1248	840
H*2	1320	2574	2178	1716	1155
I	1040	2028	1716	1352	910
J					
K	800	1560	1320	1040	700
L					
Sum	6920	13494	11418	8996	6055
Profit	17200	33540	28380	22360	15050

(c) Profit for Product P3 in five quarters

Product P4	Q1	Q2	Q3	Q4	Q5
A	135	114	87	45	15
B	225	190	145	75	25
C	630	532	406	210	70
D	135	114	87	45	15
E	990	836	638	330	110
F	1665	1406	1073	555	185
G*2	2070	1748	1334	690	230
H*2	675	570	435	225	75
I	360	304	232	120	40
J					
K					
L	675	570	435	225	75
Sum	7560	6384	4872	2520	840
Profit	24525	20710	15805	8175	2725

(d) Profit for Product P4 in five quarters

Product P5	Q1	Q2	Q3	Q4	Q5
A	69	60	66	54	57
B	483	420	462	378	399
C	138	120	132	108	114
D	276	240	264	216	228
E					
F	828	720	792	648	684
G*2	621	540	594	486	513
H*2	805	700	770	630	665
I	184	160	176	144	152
J					
K					
L	391	340	374	306	323
Sum	3795	3300	3630	2970	3135
Profit	15525	13500	14850	12150	12825

(e) Profit for Product P5 in five quarters

Table 7.1 shows the data of machine capacity (minutes per quarter) used for each of five products. The profit generated by each product in each quarter is listed in the bottom row of the five tables. Not every machine is used for processing the parts or assembling a product since there are many blanks in the above tables. For instance, Machines F, H, K and L are not used for product P1.

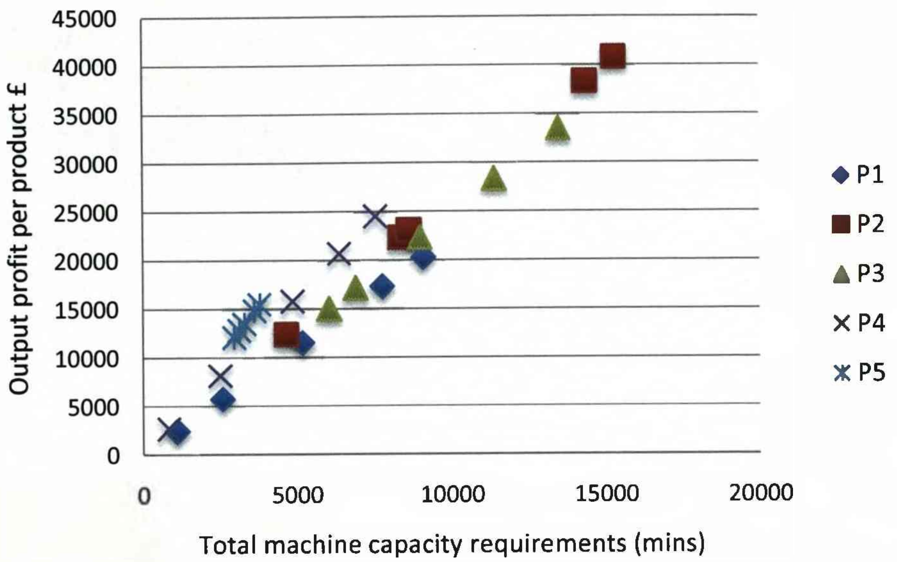


Figure 7.1: The relationship between product profit and machine capacity for the five products.

The profit for each of the five products in relation to machine capacity requirements is shown in Figure 7.1. The five points of each product shown in the graph represent the output profit of five quarters. The slopes of the data indicate that if a selection criterion is required for the product portfolio, product P1 has the least profit for a set number of machine cycle time. Product P2, although offering the most profit, requires the most machine time, especially in the last two quarters, which can be a problem if the manufacturing system does not have enough machine capacity.

The above analysis uses the total machine capacity against the production output to measure the level of profitability for the five products. The results give the decision maker another prospective to consider the whole system. However, most of the non-bottleneck machines have extra capacity which would be a good decision if reconfiguration requires utilising these capacities. The limited capacities of the bottleneck machines are major evidences when dealing with reconfiguration. For this particular reason, Table 7.2 and Figure 7.2 show the profit per bottleneck machine time for each of five products over the five quarter period.

Table 7.2: Profit per bottleneck machine time over five quarters.

Product	Q1	Q2	Q3	Q4	Q5
P1	0	3.1	2.6	2.3	4.1
P2	0	3.6	3.0	2.9	8.9
P3	35.8	3.1	2.9	3.0	8.6
P4	14.7	3.7	3.6	3.6	12.4
P5	18.8	5.1	4.7	5.0	48.2

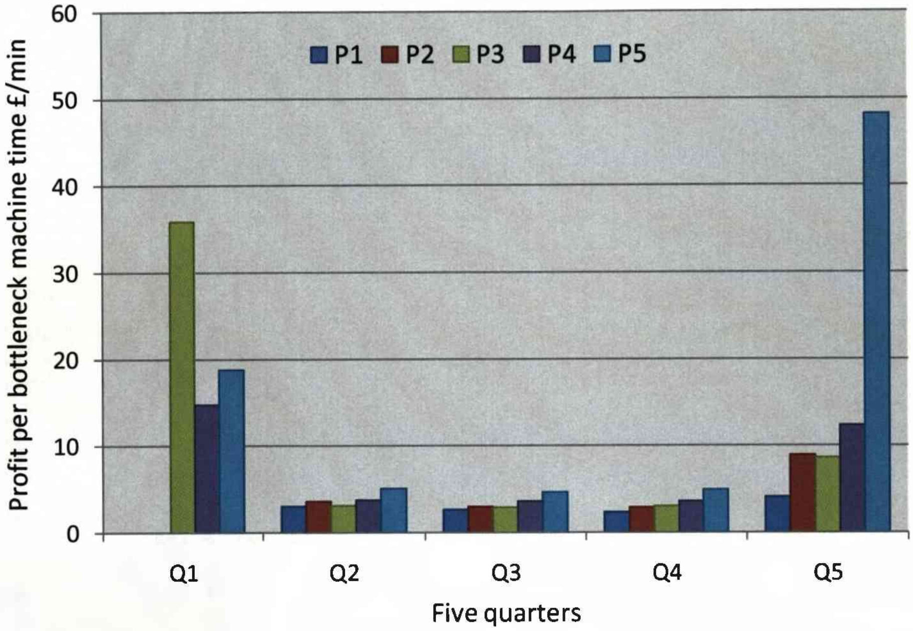


Figure 7.2: Profit per bottleneck machine time over five quarters.

With reference to Figure 7.2, products P5, P4 and P3 are obviously producing more profit per bottleneck machine time than product P1 and P2 except P1 and P2 did not use the bottleneck machine F in quarter one. As the goal of any manufacturing company is to make profit, products P3, P4 and P5 should have higher priority for in house production and products P1 or P2 should be investigated during product portfolio rearrangement.

7.3.1 Dedicated production line for P1 or P2

Building a dedicated production line for a high volume product is recognized as a standard manufacturing strategy. It is also one of the strategies which will be introduced to change product portfolio in this research. Figure 7.3 shows the demand status of the five products over the time period. Products P1 and P2 have quite high volumes compared to other products and their contributions to profits are low as investigated earlier.

The decision of building a dedicated line for product P1 or P2 is supported by the demand patterns of five products shown in the Figure 7.3. The demand for product P1 increased 1200 units over three quarters, the demand of P2 also increased by 1000 units. The manufacturing system will have to be reconfigured by a large amount to cope with this changing demand. The proposed option is to build a dedicated production line for either product P1 or P2.

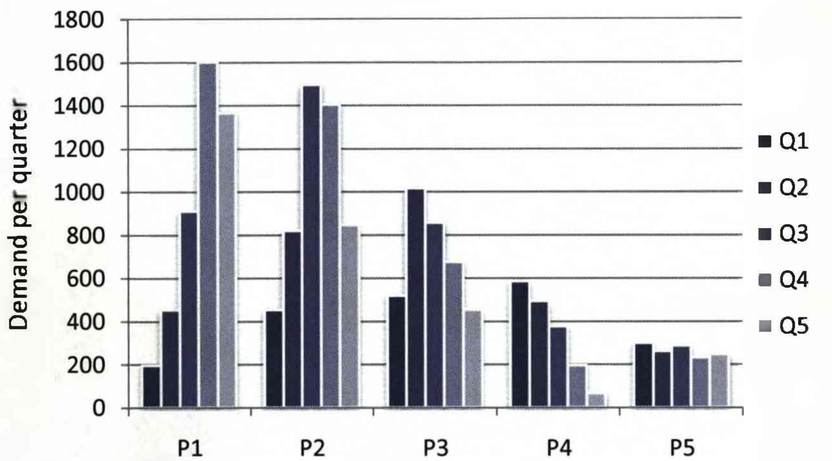


Figure 7.3: Product demand over five quarters.

In quarter one, both products P1 and P2 do not use the only bottleneck machine F for processing. Therefore, using a dedicated line for either P1 or P2 in Q1 will not reduce the utilisation of machine F. Consequently, the output of the other three products will remain the same. As discussed in Chapter six, the investment of improving machine F is very subtle under all cost models compared with the extra profit after reconfiguration. It should be noted that the product portfolio should not be changed at this stage, a reconfiguration of machine F is the easiest and the most cost effective method to reconfigure the system.

A dedicated production line for products P1 or P2 would have a major effect on the system output from the above analysis. However, simulation results show that if P1 is removed from the system, the total production time is reduced from 47169 seconds to 43075 seconds in quarter two. There are some improvements after this reconfiguration, however, when the system only has 31200 seconds working time per quarter, the output is still far from the market demand. In conclusion, there is no major effect of taking product P1 out of the manufacturing system.

As the simulation results do not match with the expectation, it is quite hard to understand at the first instance. However, it is still the best way to explain the system performance from the constraint point of view. As already discussed, machine F does not do any processing for products P1 and P2, so in this particular case, the system constraint machine F is the factor which stops the increase of system output. The following two tables give the status of the bottleneck machines after removing P1 or P2 from the production system.

Table 7.3: Machine utilisation after removing product P1.

	Q1	Q2	Q3	Q4	Q5
A					
B		6.17%	9.04%		
C		19.29%	49.21%	26.92%	
D					
E		25.71%	57.46%	32.08%	
F	23.88%	27.58%	10.71%		
G*2		4.46%	4.62%		
H*2		26.02%	54.33%	32.31%	
I		3.83%			
J					
K					
L					

Table 7.4: Machine utilisation after removing product P2.

	Q1	Q2	Q3	Q4	Q5
A					
B					
C				2.92%	
D					
E					
F	23.88%	27.58%	10.71%		
G*2					
H*2					
I		30.08%	41.00%	59.58%	24.67%
J					
K					
L					

In Table 7.3, it shows that there are still seven bottleneck machines in Q2 and six bottleneck machines in Q3 after removing product P1 from the manufacturing system. There is little justification in arranging a dedicated line for P1.

On the other hand, removing product P2 reduces the number of bottleneck machines from seven to two in Q2 and improvement in other periods as well. Simulation results show that if P2 is removed from the system, the total production time is reduced from 47169 seconds to 40827 seconds in Q2. When no overtime is allowed, the system only runs 31200 seconds per quarter. The system outputs changes before and after removing P2 out of the production system are shown in Table 7.5.

Table 7.5: The system output after removing P2
(where red indicates market demand is not met).

		P1	P2	P3	P4	P5
Q1	With P2	195	455	520	383	299
	Without P2	195		520	383	299
	Demand	195	455	520	585	299
Q2	With P2	199	567	699	363	260
	Without P2	440		699	363	260
	Demand	455	819	1014	494	260
Q3	With P2	221	581	669	347	286
	Without P2	479		669	347	286
	Demand	910	1495	858	377	286
Q4	With P2	273	714	676	195	234
	Without P2	562		676	195	234
	Demand	1599	1404	676	195	234
Q5	With P2	434	845	455	65	247
	Without P2	933		455	65	247
	Demand	1365	845	455	65	247

In Table 7.5, the red numbers indicate that the system throughput of a particular product is still below market demand. It is obvious that the machine capacity released by removing product P2 is used for producing products P1 and P1 only. The throughput of P1 is more than doubled from quarter two to quarter five although it still cannot meet the increasing market demand.

As illustrated from the above analysis, removing one product from the production system will not completely solve the reconfiguration problem. The system still cannot meet the full market demand. This method will have to be supplemented with the capacity increase of particular bottleneck machines.

7.3.2 Outsourcing parts

The manufacturing system has more than one constraint. Some constraints require more than 50% of extra capacity. It is suggested that substantial improvement to reduce machine cycle time is difficult to achieve. It also brings high investment risk with the ever-changing global market demand.

Outsourcing some parts to reduce the capacity requirement on bottleneck machine is a viable manufacturing strategy. Simulation can produce useful information to support the decision on the choice of parts for outsourcing and how many. The following study is carried out based on the same simulation model.

There are a lot of factors to consider when one decides which parts to outsource. The supply conditions such as price, lead time and quality are all essential for the decision makers. This theoretical study will not consider all aspects of outsourcing but focus on the impact of outsourcing to production. Analysis based on simulation data and relevant outsourcing principles are presented in this section.

Table 7.6: Machine time for twenty parts.

Machine / Parts	A	B	C	D	E	F	G	H	I
RM1	3		6						
RM2		6	6						
RM3				3			8		8
RM4	3		14			12			8
RM5		5					17	17	
RM6		6	6		10		8	18	
RM7	4			7	11		10	17	
RM8							10		
RM9		5			11			18	10
RM10	4	6	6		10				10
RM11	3	5		4					8
RM12						12		18	
RM13				3	11	13	18		
RM14	5		7	3					
RM15		5		11		24			
RM16							10	15	
RM17	3	10	7						
RM18		5			11	12	8		
RM19		5		4		12	8	15	8
RM20			6	4			8		

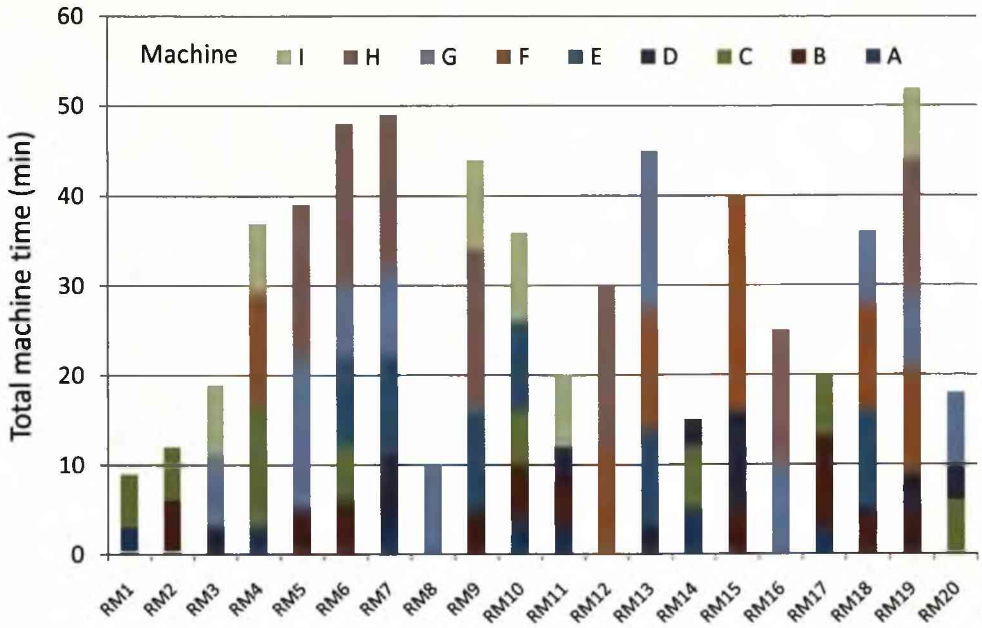


Figure 7.4: Machine cycle time for twenty parts.

Table 7.6 and Figure 7.4 show the amount of time for nine single process machines (A to I) take to produce each of the twenty parts. The diagram shows the machine cycle time for each part. It is obvious that outsourcing RM19 will release the most machine capacity. However, the goal is not to reduce the total machine time; but to reduce the bottleneck machine time. The information here does not take the market demand into consideration as well. Therefore, quarter two situation is used in the following analysis.

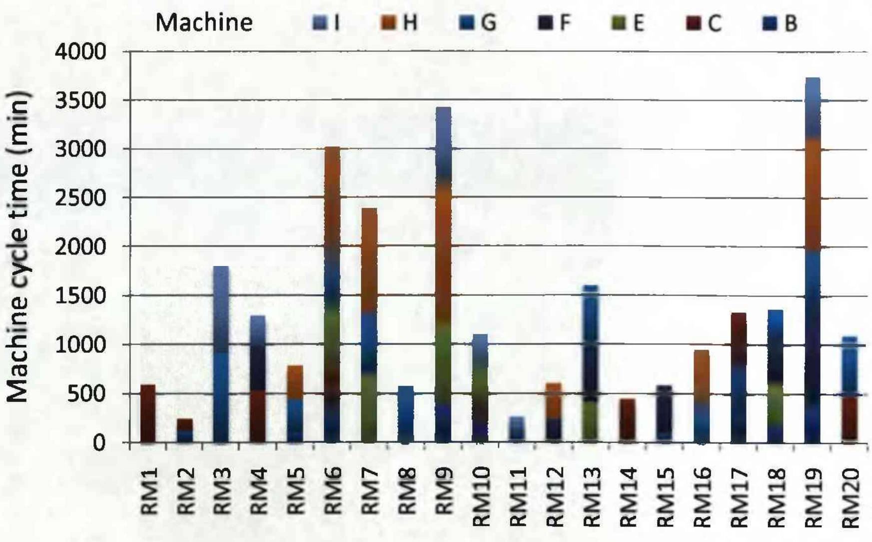


Figure 7.5: Capacity requirement for twenty parts on bottleneck machines.

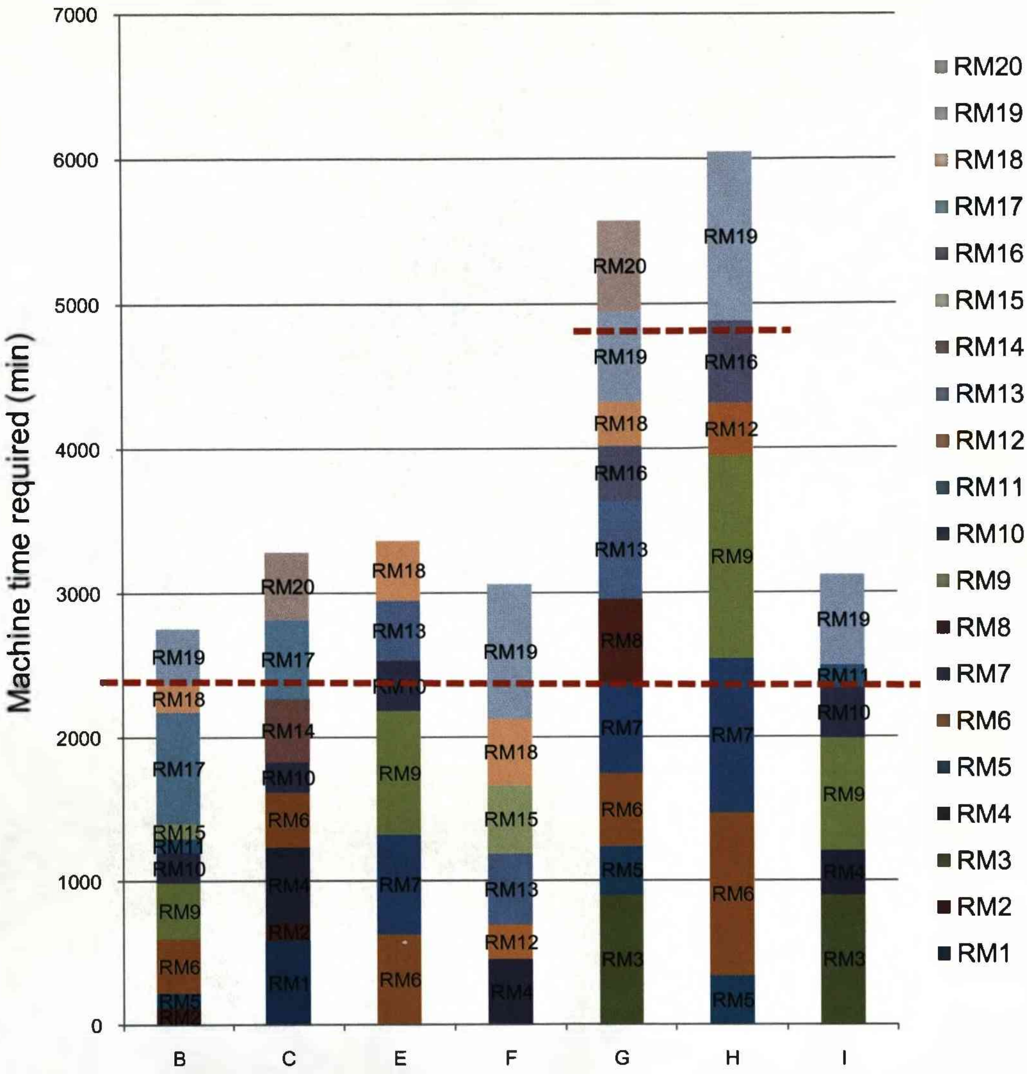


Figure 7.6: Bottleneck machines' capacity utilisation for twenty parts.

Generally speaking, outsourcing should be kept as low as possible as it will affect the effort of the purchasing department and keep most of the production in-house. As shown in Figure 7.6, it is obvious that RM19 should be an outsourcing part as it takes much of the capacity from five bottleneck machines out of seven. The red dashes represent the limited capacity per bottleneck per week. There are two machines G and H, hence the available capacity for these two machine types are doubled to 4800 minutes which is indicated by the red dash lines. Outsourcing RM6 and RM10 can also reduce multiple bottleneck machine utilisations as shown in Figure 7.7.

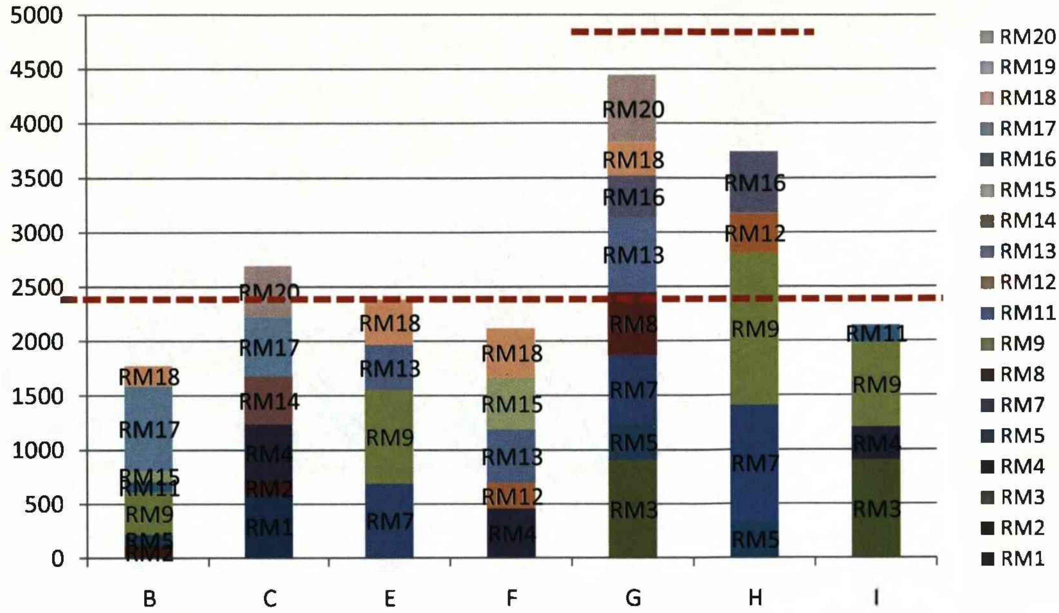


Figure 7.7: Bottleneck machines' capacity utilisation after outsourcing

After outsourcing three parts, there is only Machine C slightly above the capacity limit, in other word, there is only one bottleneck machine. For the rest of products, partial outsourcing should be considered so that there is no further reduction of non-bottleneck machines utilisation. Bottleneck machines B, F, G, H and I have dropped the capacity requirement way below the limit, which gives more flexibility during production scheduling. Another criterion that should be used regarding outsourcing is maximizing machine utilisations. The problem of using this criterion is that the machine utilisation is normally lower than the capacity requirement even when the machine has extra idle capacity to be utilised. This is due to the scheduling problem, machines often have to wait for parts to arrive from previous machines.

Disregarding the difficulty of purchasing varies parts, the easiest way to make an outsourcing decision is to use simulation. Simply purchase all the parts that the simulation results inform that the system could not make within a certain demand period. Table 7.7 shows the simulation results of how many parts have to be outsourced since the system cannot produce sufficiently to meet the market demand in quarter two. There are 15 parts out of 20 parts that do not meet the demand, given that this represents three quarters of the parts, it is unlikely for them to be outsourced. Therefore the outsourcing decision will have to be made for certain parts and release the capacity on bottleneck machines so they can satisfy more market demand.

Table 7.7: Simulation result for outsourcing.

	Demand	Outsourcing
RM1	98	38
RM2	20	
RM3	113	42
RM4	38	9
RM5	20	
RM6	63	19
RM7	63	19
RM8	58	9
RM9	78	22
RM10	35	20
RM11	20	
RM12	20	
RM13	38	9
RM14	63	19
RM15	20	
RM16	38	9
RM17	78	22
RM18	38	9
RM19	78	22
RM20	78	22

Two product portfolio restructuring strategies are investigated and the simulation results are presented in this chapter. The results of the simulation study indicate that one step reconfiguration could not solve the production problem for the simulated manufacturing system. Due to the degree of complexity and fast increasing market demand, a mixed decision of either dedicated line with machine improvement or outsourcing with the machine improvement would be more appropriate for the situation much better than just one reconfiguration method.

Chapter 8
Conclusion and Future Research

8.1 Discussion

The choice of manufacturing reconfiguration is simple to describe but difficult to make. As mentioned in the research context (Section 1.2), reconfiguration is only one of many options that are available to manufacturing businesses for staying competitive in this global market. Individual companies can pursue the goals and meet the market demand by system or machine level capacity increase when the manufacturing reconfiguration is desirable. However, other methods can be taken to solve the capacity problem when the reconfiguration is necessary such as addition work shifts, extending working hours, carrying extra stock, sub-contracting or building extra manufacturing plant. This research has developed a systematic methodology based on simulation and optimisation to guide the decision making during the reconfiguration, focusing on increasing the production capacity.

The following results and conclusions summarise the research and outline the contribution of this research project.

This research not only produces new insights that will be outlined in this chapter, but also has demonstrated the process of simulating a medium sized, multi-stage multi-product manufacturing system. Computer simulation is essential in order to meet the research objectives. Only necessary information is embedded in the model which minimizes simulation time. With some reasonable assumptions, the model generates system performance information efficiently.

A method of using 'product life cycle' is developed to predict near future market demand. Market demand is critical information for the reconfiguration of manufacturing system. The product life cycle trend indicates a positive or negative demand trend. Aggregating the demand trend for a family of products could provide machine capacity requirement information. Varying the product types and increasing the number of products in the product family will reduce the turbulence of the total demand as the increase and decline between products may neutralise the life cycle effect. With dynamic product life cycles, the shorter the product life cycle, the more important the role it plays to predict market demand.

The Theory of Constraint is employed in the research for the identification and quantification of bottleneck machines. Comparing the required machine capacity against market demand with the available capacity is essential for quantifying the amount of reconfiguration for the bottleneck machine. In the simulation model, the processing of twenty different parts on multiple machines are required before final assembly into five products. There are various bottleneck machines in all five demand periods due to the degree of complexity of the system.

This research proved and concluded that the machine capacity requirement calculated based on Theory of Constraint does not match the actual requirement in order to meet the market demand. Production scheduling is a very important factor for manufacturing output. The more complex the system, the more difficult it is to achieve 100% machine utilisation. It is shown in this research that over provision of the machine capacity would lead to a reduction of system output under the same scheduling rule. As this result is unexpected from the Theory and Constraints point of view and difficult to explain and demonstrate in the 5P/12M model, a small scale 2P/3M is built in the same mechanism of the bigger model to illustrate this particular scenario. It is explained in Section 4.7 that the output decrease is due to schedule rule. Therefore, scheduling should always be controlled and analysed along with system reconfigurations. Two scheduling rules are studied in this research, i.e. TOC and SPT. The results suggest that different rules favour different aspects of system output. In this case, the TOC based scheduling rule gives better system profit while the Shortest Processing Time improves the system throughput.

Increasing or adding more capacity of bottleneck machines during manufacturing reconfiguration would normally lead to the increase of the system output although this is not always true due to the impact of other system parameters such as scheduling. When the system throughput increases after adding more bottleneck machine capacity, it clearly impacts on the utilisation of the non-bottleneck machines. The research results show that a non-bottleneck machine could become a bottleneck machine during the investigation and it is important to take that into consideration when making reconfiguration decisions.

The optimisation study demonstrates that the simulated annealing is an efficient algorithm and gives better results than other five available algorithms in WITNESS. Comparison of results from hill climb and simulated annealing demonstrated the difference between local optimisation and global optimisation. The optimisation search also produced an interesting result that the WIP should be three for the best system output for most of the experiment settings as the process balance was the best with this WIP level. However, it is only applicable to this particular simulation manufacturing system. It is suggested that the best WIP level can be obtained by optimisation search. There certainly has an impact on the step size chosen for optimisation parameters. However, the research concluded that such impact is not highly significant.

This research established a cost model for manufacturing reconfiguration which was based on the machine capacity reconfiguration data. Hence, it connects the cost function with the simulation research and could provide other important information to the decision makers. Nine cost models were generated with random numbers associated with the machine costs. These models gave an exemplary indication of reconfiguration cost and real life data could also feed into the model to generate more accurate configuration cost for practical application.

The product portfolio analysis provided another reconfiguration solution in addition to the traditional machine capacity configurations. With shorter product life cycle, a family of products and the production system would have to change along with the market expectations. Product portfolio analysis gives the decision makers information support on how to change the structure of the product family. The use of dedicated production line and part outsourcing are studied in the Chapter 7 and both methods were demonstrated to be viable.

8.2 Conclusions

When planning for the reconfiguration of manufacturing system to cope with product demand over a planning horizon, manufacturers usually face an important decision regarding how to select the optimal capacities. This research aims at design and construct a methodology to optimize the system reconfiguration, focusing on the capacity reconfiguration.

The primary research objectives are achieved. A simulation model for a medium size manufacturing system is designed and constructed to serve the purpose of the research. Machine reconfiguration options are investigated and established in the simulation model. The reconfiguration optimisation is carried out using several optimisation algorithms based on the simulation model. A cost function has been built to analysis the hard and soft cost during reconfiguration. Product portfolio restructuring is studied and several options are investigated during the research.

This research provides a simulation based methodology to formulate the optimal reconfiguration of manufacturing systems for meeting new demands. The methodology proposes a step by step simulation, reconfiguration and optimisation approach to address the capacity increase problem. The research is focused on generating and analyzing simulation results for reconfiguration and optimisation. The results represent a sound basis for making informed reconfiguration and planning decisions and for prioritizing alternative solutions. This methodology has the ability to integrate and quantify the structural, operational and scheduling-related decision-making characteristics of manufacturing systems to support the reconfiguration decision-making.

8.3 Future research

The research tackled the manufacturing reconfiguration which can be implemented on most manufacturing systems. It has opened up a wide range of research on manufacturing system reconfiguration. The following recommendations are therefore made for future research:

1. Apply the reconfiguration mechanism proposed in this research to a real life manufacturing system. Analysis of a real manufacturing system would generate new insights on the improvement of the reconfiguration methodology.
2. The research results are based on one particular model constrained by certain assumptions and generated with limited data. The results could be considered as providing a generalised set of conclusions. In order to explore and confirm these conclusions, it would be necessary to test on other types of

manufacturing systems such as flow-shop manufacturing, cellular manufacturing and mass production manufacturing.

3. Optimisation search on the reconfiguration of manufacturing system does not guarantee the result is indeed the optimum. Other available optimisation algorithms such as genetic algorithm, ant colony may improve the optimisation results. It is necessary to establish a method to prove the quality of optimisation results.
4. The research on reconfiguration cost discussed the cost of machine capacity changes in detail. However, the associated system and administration cost during reconfiguration is also important and a more comprehensive cost analysis on cost associated with machine capacity change should be set up in order to achieve more accurate costing.
5. More product portfolio information such as the competitiveness of certain product, the forecast demand, potential new market etc. should be taken into consideration in order to provide more options for product portfolio restructuring.

References

Aldowaisan, T., & Allahverdi, A. (2003). New heuristics for no-wait flowshops to minimize makespan. *Computer Operation*, *30*, 1219-1231.

Alexander, C. W. (2006). Discrete event simulation for batch processing. *Winter Simulation Conference*, (pp. 1929-1934).

Aly, A. a. (1993). Design of an FMS decision support system. *International Journal of Production Research*, *31* (10), 2257-2273.

Aytuğ, H., & Doğan, C. A. (1998). A framework and a simulation generator for kanban-controlled manufacturing systems. *Computers and Industrial Engineering*, *34* (n2), 337-350.

Baid, N. a. (1994). Integrated decision support systems for FMS: using intelligent simulation. *International Journal of Production Research*, *32* (4), 951-965.

Ball, P. (2001). *Introduction to discrete event simulation.* Retrieved from http://www.dmem.strath.ac.uk/~pball/simulation/simulate.html

Banks, J. (1999). Discrete Event Simulation. *Winter Simulation Conference*, (pp. 7-13).

Banks, J., Aviles, E., Mclaughlin, J., & Yuan, R. (1991). The simulator: new member of the simulation family. *Interfaces*, *21* (n2), 76-86.

Banks, J., Carson, J., Nelson, B., & Nicol, D. M. (2000). *Discrete Event System Simulation* (3rd Edition ed.). Upper Saddle Rive, New Jersey: Prentice-Hall.

Banks, J., Hugan, J. C., Lendermann, P., McLean, C., Page, E. H., Pegden, C. D., et al. (2003). Future of the simulation industry. *Winter Simulation Conference*, (pp. 2033 - 2043).

Baptiste, P. L. (2001). *Constraint-Based Scheduling* (Vol. 39). Springer.

Barton, R. R., Fishwick, P. A., Henriksen, J. O., Sargent, R. G., & Twomey, J. M. (2003). simulation - past, present and future. *Winter Simulation Conference*, (pp. 2044 - 2050).

Baykasoglu, A., & N.N.Z, G. (2001). A simulated annealing algorithm for dynamic layout problem. *Computers & Operations Research* , *28*, 1403-1426.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)* , *35* (3), 268-308.

Bradford, J., & Childe, S. (2002). A non-linear redesign methodology for manufacturing systems in SMEs. *Computers in Industry* , *49* (1), 9-23.

Browne, J. D. (1984). Classification of Flexible Manufacturing Systems. *FMS Magazine* , *2* (2), 114-117.

Bryan, A. H. (2007). Concurrent product portfolio planning and mixed product assembly line balancing. *CHINESE JOURNAL OF MECHANICAL ENGINEERING -ENGLISH EDITION* , *20* (1), 96-99.

Burke, E. K., & Kendall, G. (2005). *Search Methodologies: Introductory tutoriasl in optimization and decision support techniques.* New York: Springer.

Byrne, M. a. (1997). Real-time operational control of an FMS with full routing flexibility. *International Journal of Production Economics* , *51* (nl/2), 109-113.

Cox, J. F. (1992). *APICS Dictionary.* Falls Church, Virginia: American Production and Inventory Control Society.

Deif A. M., E. W. (2005). Scalable Capacity in Reconfigurable Manufacturing Systems. *CIRP 3rd International Conference on Reconfigurable Manufacturing* (pp. 10-12). Ann Arbor MI: University of Michigan .

Deif, A. M., & Elmaraghy, W. (2007). Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system. *International Journal of Advanced Manufacturing Technology* , *32* (n 5-6), 557-562.

Dorigo, M., & Colorni, V. M. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics* , *26* (1), 29-41.

Driva, H. P. (2000). Measuring Product Development Performance in Manufacturing Organisations. *International Journal of Production Economics*, *63* (2), 147-159.

ElMaraghy, A. M. (2007). Investigating optimal capacity scalability scheduling in a reconfigurable manufacturing system. *The International Journal of Advanced Manufacturing Technology*, *32* (5-6), 557-562.

Engineering Research Center for Reconfigurable Machining Systems. (2001, 03 26). Retrieved 2008, from http://www.nsf.gov/pubs/2000/nsf00137/nsf00137l.htm

Faget, P., Eriksson, U., & Herrmann, F. (2005). Applying discrete event simulation and an automated bottleneck analysis as an aid to detect running production constraints. *Winter Simulation Conference*, (pp. 1401-1407).

Ferrell, W., Sale, J., Sams, J., & Yellamraju, M. (2000). Evaluating simple scheduling rules in a mixed shop environment. *Computers & Industrial Engineering*, *38* (1), 39-66.

Finch, R. L. (1992). Theory of constraints and linear programming: a comparison. *International Journal of Production Research*, *30* (6), 1471-1478.

Fulkerson, B. (1997). A Response to Dynamic Change in the Market Place. *Journal of Decision Support Systems*, *21*, 199-214.

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: W.H.Freeman.

Goldratt E, M. (1990). *The Haystack Syndrome.* Croton-on-Hudson: NY: North River Press.

Group, L. (n.d.). *LANNER the value of knowing*. Retrieved from www.lanner.com

Group, L. (2003-2006). *WITNESS Optimizer.* Retrieved from http://www.lanner.com/products/simulation_suite/witness_optimizer.php

Harmonosky, C. a. (1995). Investigating the application potential of simulation to real-time control decisions. *International Journal of Computer Integrated Manfacturing*, *8* (2), 126-132.

Harmonosky, C. (1995). Simulation-based real-time scheduling: review of recent developments. *Winter Simulation Conference*, (pp. 220-225).

Hlupic, V. (2000). Simulation software: an Operational Research Society survey of academic and industrial users. *Winter Simulation Conference*, (pp. 1676 - 1683).

Hlupic, V. & Paul, R. (1995). A critical evaluation of four manufacturing simulators. *International Journal of Production Research* , *33* (n10), 2757-66.

Hon, K.K.B. & Xu, S. (2007). Impact of product life cycle on manufacturing systems reconfiguration. *CIRP annals* , *56* (1), 455-458.

Hu, S. (2005). Paradigms of manufacturing - a panel discussion. *3rd Conference on Reconfiguable Manufacturing*. Ann Arbor, Michigan, USA.

ISKANDER, S. S. (1977). A Survey of Scheduling Rules. *Operations Research* , *25* (1).

Iyer, A. a. (1998). Modeling and simulating operating policies for manufacturing cells. *IIE Transactions* , *30* (9), 785-794.

J Antony, M. K. (2008). Gearing Six Sigma into UK manufacturing SMEs: results from a pilot study. *Journal of the Operational Research Society* , *59* (4), 482-493.

Johansson, B., & Kaiser, J. (2002). turn lost production into profit -- discrete event simulation applied on resetting performance in manufacturing systems. *Winter Simulation Conference* , (pp. 1065-1072).

Johansson, B., Johnsson, J., & Kinnander, A. (2003). Information structure to support discrete event simulation in manufacturing systems. *Winter Simulation Conference*, (pp. 1290-1295).

Jun Du, Y.-Y. J. (2006). A real-option approach to flexibility planning in reconfigurable manufacturing system. *International Journal of Advanced Manufacturing Technology* , *28* (11-12), 1202-1210.

Kelton, A. M. (2000). *Simulation modeling and analysis*. McGraw-Hill.

Kim JU, K. Y. (2002). Heuristic algorithms for a multi-period multi-stop transportation planning problem. *Journal of the Operational Research Society* , *53*, 1027-1037.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science* , *220* (4598), 671-680.

Koren, Y. (2006). *Reconfigurable Manufacturing Systems and Transformable Factories.* Springer Berlin Heidelberg.

Koren, Y., & Kota, S. (1999). *Patent No. 5,943,750.* US.

Koren, Y., Jovane, F., Heisel, U., Moriwaki, T., Pritschow, G., Ulsoy, G., et al. (1999). Reconfigurable Manufacturing Systems. *CIRP Annals* , *48* (2), 6-12.

Laarhoven, P., Aarts, E., & Lenstra, J. (1992). Job shop scheduling by simulated annealing. *Operations Research* , *40* (1).

Landers, R. G. (2001). Reconfigurable machine tools. *CIRP Annual* , *50*, 269-274.

Laporte, I. a. (1996). Metaheuristics: A bibliography. *Annals of Operations Research* , *63*, 513-623.

Levitt, T. (1965). Exploit the product life cycle. *Harvard Business Review* , *43*, 81-94.

Luke, B. T. (2003). *Simulated annealing.* Retrieved 2007, from http://fconyx.ncifcrf.gov/~lukeb/simann1.html

Luss, H. (1982). Operations research and capacity expansion problems: a survey. *Operation research* , *30* (5), 907-947.

M A Youssff, H. A. (2006). Modelling and optimization of multipleaspect RMS configuration. *International Journal of Production Research* , *44* (22), 4929-4958.

M.A.Youssef, A., & A.Elmaraghy, H. (2006). Assessment of manufacturing systems reconfiguration smoothness. *International Journal of Advance Manufacturing Technology* , *30* (1-2), 174-193.

Manne, A. (1961). Capacity expansion and probabilistic growth. *Econometrica , 29* (4), 632-649.

Mansurov, N. P. (2001). Improving Time-To-Marketing Using SDL Tools and Techniques. *Computer Network , 35* (6), 667-691.

Mehrabi, M. U. (2000). Reconfigurable Manufacturing Systems: Key to Future Manufacturing. *Journal of Intelligent Manufacturing , 11* (4), 403-419.

Meller, R., & Bozer, Y. (1996). A new simulated annealing algorithm facility layout problem. *International Journal of Product Research , 34*, 1675-1692.

Murphy, C., & Perera, T. D. (2001). The definition and potential role of simulation within an aerospace company. *Winter Simulation Conference*, (pp. 829-827).

Nandkeolyar, U., Ahmed, M. U., & and Pai, A. R. (1998). A simulation study of a manufacturing cell application. *Production and Inventory Management Journal , 39* (2), 29-37.

Ng, W., Elmirghani, J., & Broom, S. (2001). Optimum variable step-size adaptation in a speech driven in-service non-intrusive measurement device. *IEEE International Conference, 3*, pp. 925 - 929.

Ohno, T. (1988). *Toyota Production System - beyond large-scale production.* Productivity Press.

Park, Y. M. (1998). Simulation and analysis of the Mercedes-Benz All activity Vehicle (AAV) production facility . *Winter Simulation Conference*, (pp. 921-926).

Parunak, H. (1991). Characterising the Manufacturing Scheduling Problem. *Journal of Manufacturing Systems , 10* (3), 241-258.

Patrick, S., & Carlo, H. J. (2007). Integrating reconfiguration cost into the design of multi-period scalable reconfigurable manufacturing systems. *Journal of Manufacturing Science and Engineering , 129* (n1), 202-210.

Pierre Chardaire, J. L. (1995). Thermostatistical persistency. A powerful improving concept for simulated annealing algorithms. *European Journal of Operational Research , 86*, 565-579.

R. Sivasubramanian, V. S. (2003). Optimum production schedule and profit maximisation using the concept: theory of constraints. *International Journal of Manufacturing Technology and Management*, *5* (4), 325-337.

S. Voß, S. M. (1999). *Meta-Heuristics - Advances and Trends in Local Search Paradigms for Optimization.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

S.Y, S. (2000). *Design Principles and Methodologies for Reconfigurable Manchining Systems* (Ph.D dissertation ed.). Ann Arbor , MI: The University of Michigan.

Saad, S. M. (2003). The reconfiguration issues in manufacturing systems. *Journal of Materials Processing Technology*, *138*, 277-283.

Saad, S. M., Baykasoglu, A., & Gindy, N. (2002). An integrated framework for reconfiguration of cellular manufacturing systems using virtual cells. *Production Planning and Control*, *13* (4), 381-393.

SAAD, S. M., BAYKASOGLU, A., & GINDY, N. N. (2002). A new integrated system for loading and scheduling in cellular manufacturing. *International Journal of computer intergrated manufacturing*, *15* (1), 37-49.

Saad, S., & Gindy, N. (1999). Handling internal and external distribution in responsive manufacturing enviroments. *Production planning and control* , *9* (8), 760-770.

Savory, P. (1991). Flexible facility layout by simulation. *Computers & Industrial Engineering*, *20* (1), 155-166.

Schriber, T. a. (2001). Inside discrete-event simulation software: how it works and why it matters. *Winter Simulation Conference*, (pp. 158-167).

Setchi, R. M., & Lagos, N. (2004). Reconfigurability and reconfigurable manufacturing systems: state-of-the-art review. *Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on*, (pp. 529-535).

Sethi, A. a. (1990). Flexibility in Manufacturing: A Survey. *International Journal Flexible Manufacturing System*, *2* (4), 289-328.

Shabaka AI, E. H. (2004). Structural mapping between operation clusters and machine configuration for RMS. *International Workshop on Advanced Manufacturing Technologies and Integrated Systems AMT* (pp. 1-2). London: Ontario.

Smith, J. J. (1994). Discrete event simulation for shop floor control. *Winter Simulation Conference*, (pp. 962-969).

Spicer, P. K.-H. (2002). Design principles for machining system configurations. *CIRP Annual, 51*, 275.

Spicer, P. K.-h. (2005). Scalable Reconfigurable Equipment Design Principles. *International Journal of Production Research , 43* (22), 4839-4852.

Sridhar, J., & Rajendran, C. (1993). Scheduling in a cellular manufacturing system: a simulated annealing approach. *International Journal of Product Research , 31*, 2927-2945.

Steiglitz, C. H. (1982). *Combinatorial Optimization - Algorithms and Complexity.* New York: Dover Publications Inc.

Stein R.E. (2003). *Re-Engineering the Manufacturing System* (2nd edn ed.). New York: Marcel Dekker.

Suresh, G., & Sahu, S. (1994). Stochastic assembly line balancing using simulated annealing. *International Journal of Product Research , 32*, 1801-1810.

Swain, J. (2005). *ORMS a publication of informs.* Retrieved from www.lionhrtpub.com/orms/surveys/simulation

Swain, J. (2001). Power Tools for Visualization and Decision Making: 2001 Simulation Software Survey. *OR/MS Today , 28* (1), 53-63.

Swisher, J. R., Hyden, P. D., Jacobson, S. H., & Schruben, L. W. (2000). A survey of simulation optimization techniques and procedures. *Winter Simulation Conference*, (pp. 119 - 128).

T.Li, Y. M. (2006). Computer-aided reconfigurable planning: an artificial intelligence-based approach. *Journal of computing and information science in engineering*, *6* (9), 230-240.

Taj, S., Cochran, D., & Duda, J. l. (1998). Simulation and production planning for manfuacturing cells. *Winter Simulation Conference*, (pp. 973-978).

Tan, W., & Khoshnevis, B. (2000). Integration of process planning and scheduling- a review. *Journal of Intelligent Manufacturing*, *11*, 51-63.

Tan, W., & Khoshnevis, B. (2000). Integration of process planning and scheduling- a review. *Journal of intelligent Manufacturing*, *11*, 51-63.

Terwiesch, C. B. (2001). Learning and Process Improvement During Production Ramp-Up. *International Journal of Production Economics*, *70* (1), 1-19.

W. S. GERE, J. (1966). Heuristics in Job Shop Scheduling. *Management Science*, *13*, 167-190.

Westkämper E, S. T.-H. (2000). Production planning and control with learning technologies: simulation and optimization of complex production processes. *In Leondes G (ed.) Knowledgebased system*, *3*.

Wight, O. W. (1984). *Production and Inventory Management in the Computer Age.* New York: Van Nostrand Reinhold Company Inc.

Williams, E. a. (1997). Pallet optimization and throughput estimation via simulation. *Winter Simulation Conference*, (pp. 744-749).

Williams, E. J. (1996). Making Simulation a Corporate Norm. *Summer Computer Simulation Conference*, (pp. 627-632).

Williams, E., & and Celik, H. (1998). Analysis of conveyor systems within automotive final assembly. *Winter Simulation Conference*, (pp. 915-920).

Wolsey, G. a. (1988). *Integer and Combinatorial Optimization.* New York: John Wiley & Sons.

www.metaheuristics.net. (2000).

Y.B, P. (1988). An evaluation of static flow-shop scheduling heuristics in dynamic flow-shop models via computer simulation. *14* (2), 103 - 112 .

Y.Koren, & Weber, S. H. (1998). Impact of Manufacturing System configuration on Performance. *CIRP annual , 47* (1), 369-378.

Youssef, A. a. (2005). A New Approach for RMS Configuration Selection. *CIRP 3rd International Conference on Reconfigurable Manufacturing*, (pp. 10-12). Ann Arbor MI.

Bibliography

Banks, J., ed. (1998), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, John Wiley & Sons.

Averill M. Law., David M. Kelton., (2000), *Simulation Modelling and Analysis, 3rd edition,* McGraw-Hill Higher Education.

Shen, Weiming. "Distributed Manufacturing Scheduling Using Intelligent Agents." *IEEE Intelligent Systems* 17, no. 1 (2002): 88-94.

Goldratt,E.M., 1984 The Goal(Croton-on-Hudson, NY:North River Press).

Goldratt,E.M., 1990a, Theory of Constraints(Croton-on-Hudson, NY:North River Press).

Goldratt,E.M., 1990b, The Haystack Syndrome (Croton-on-Hudson, NY:North River Press).

Fox, Robert; Goldratt, Eliyahu M. (1986). The race. (Croton-on-Hudson, NY: North River Press).

Appendix A
Scheduling rule simulation results

Table A.1: Throughput changes with scheduling rules and machine capacity.

Total throughput	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
TOC based scheduling	1853	2087	2102	2092	2046
Shortest processing time	1782	2165	2253	2247	2207

TOC based scheduling (0% excess capacity)	1983	2776	3396	3291	2518
Shortest processing time(0% excess capacity)	2047	2717	3509	3523	2531

TOC based scheduling (5% excess capacity)	1959	2876	3503	3456	2571
Shortest processing time(5% excess capacity)	2047	2766	3565	3656	2588

TOC based scheduling (10% excess capacity)	1955	2856	3603	3593	2625
Shortest processing time(10% excess capacity)	2054	2811	3643	3799	2629

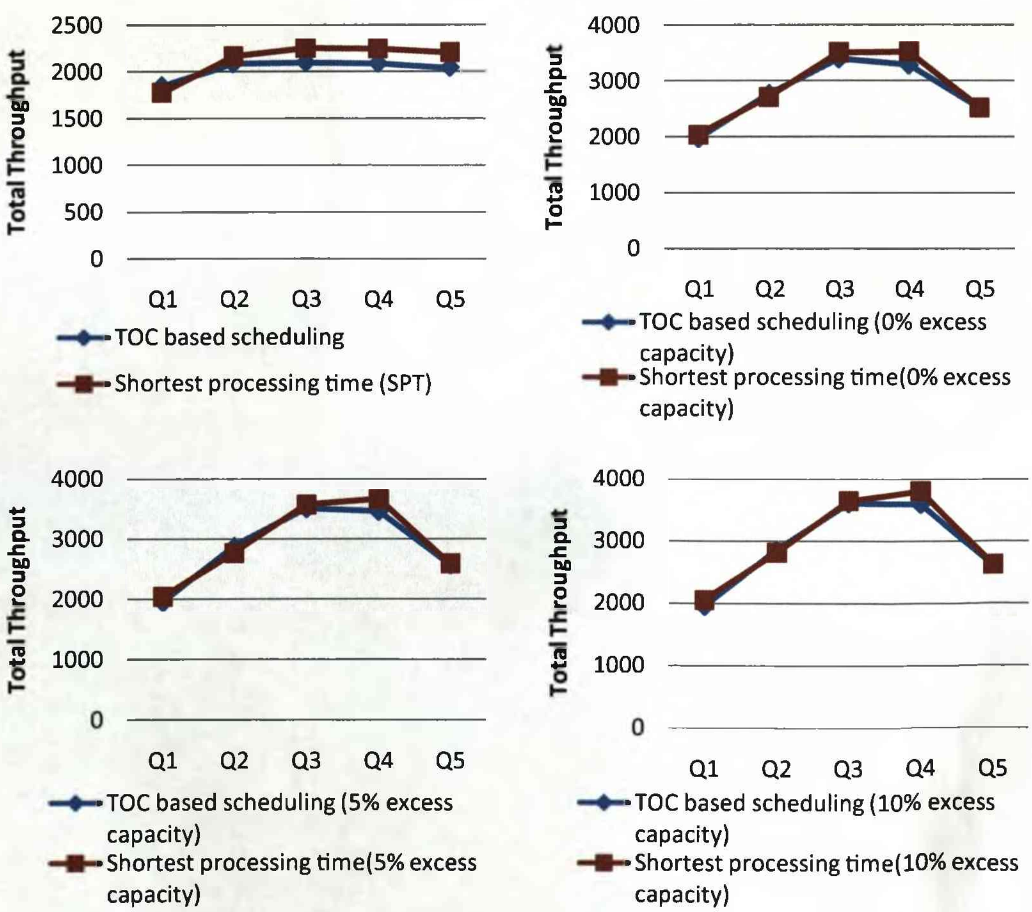
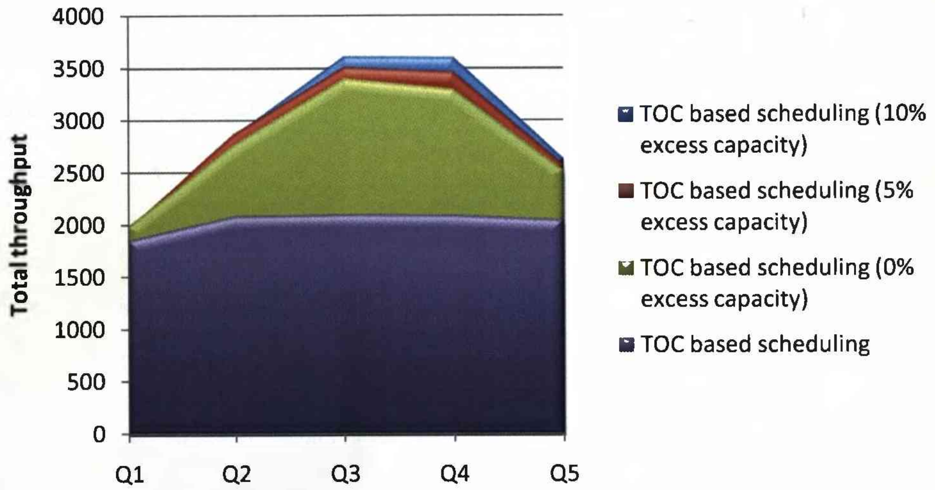


Figure A.1: Throughput changes with scheduling rules and machine capacity.

Table A. 2: Throughput changes by increasing machine capacity under two scheduling rules.

Total throughput	Q1	Q2	Q3	Q4	Q5
TOC based scheduling	1853	2087	2102	2092	2046
TOC based scheduling (0% excess capacity)	1983	2776	3396	3291	2518
TOC based scheduling (5% excess capacity)	1959	2876	3503	3456	2571
TOC based scheduling (10% excess capacity)	1955	2856	3603	3593	2625

Shortest processing time	1782	2165	2253	2247	2207
Shortest processing time(0% excess capacity)	2047	2717	3509	3523	2531
Shortest processing time(5% excess capacity)	2047	2766	3565	3656	2588
Shortest processing time(10% excess capacity)	2054	2811	3643	3799	2629



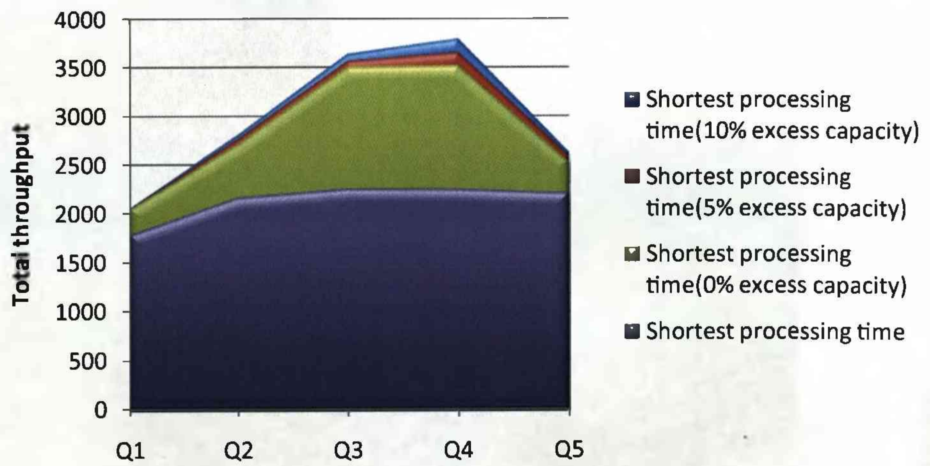


Figure A.2: Throughput changes by increasing machine capacity under two scheduling rules.

Table A.3: Profit changes with scheduling rules and machine capacity.

Total profit (£)	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*
TOC based scheduling	828405	907595	912075	853420	769385
Shortest processing time	801555	878025	880595	834095	775240

TOC based scheduling (0% excess capacity)	905715	1138715	1288615	1126495	847265
Shortest processing time(0% excess capacity)	934135	1106820	1280195	1185285	849410

TOC based scheduling (5% excess capacity)	908785	1177290	1318805	1172910	856010
Shortest processing time(5% excess capacity)	934135	1127890	1303450	1219540	858815

TOC based scheduling (10% excess capacity)	917815	1190440	1355335	1209955	864920
Shortest processing time(10% excess capacity)	937950	1147240	1334740	1259130	865580

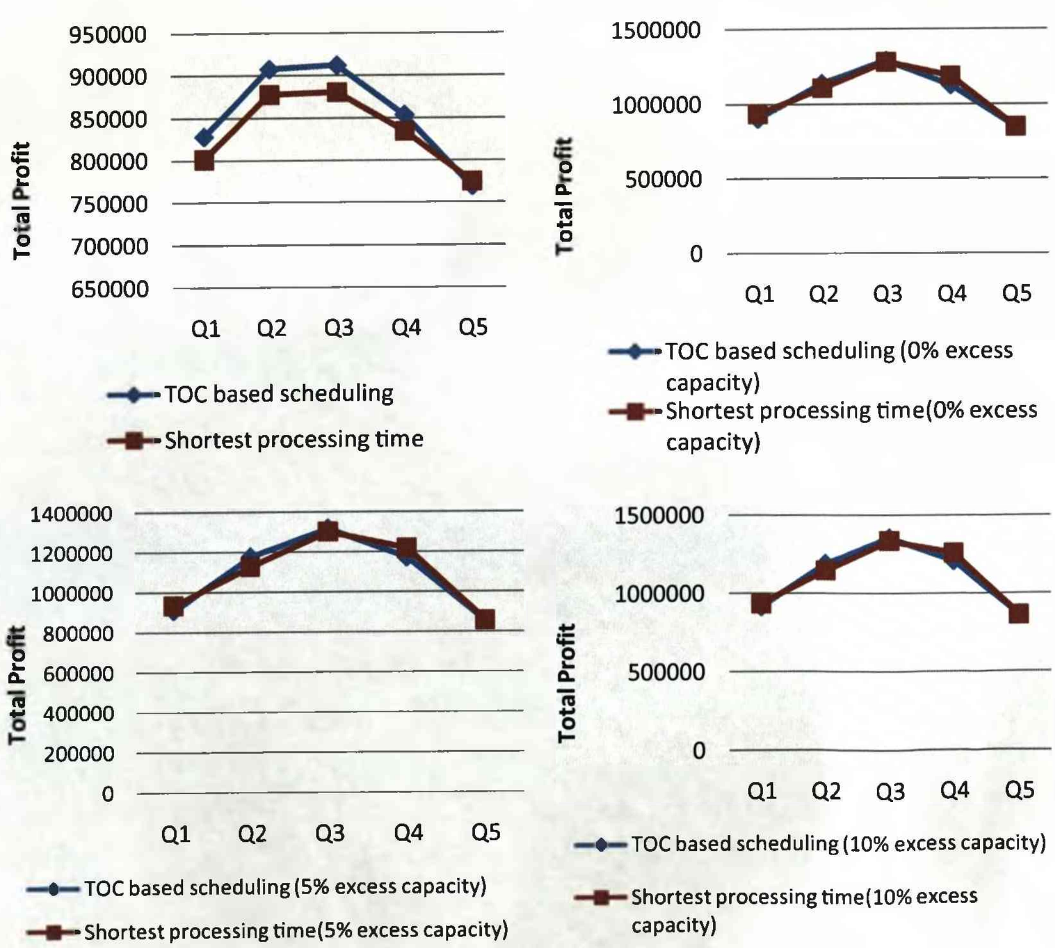
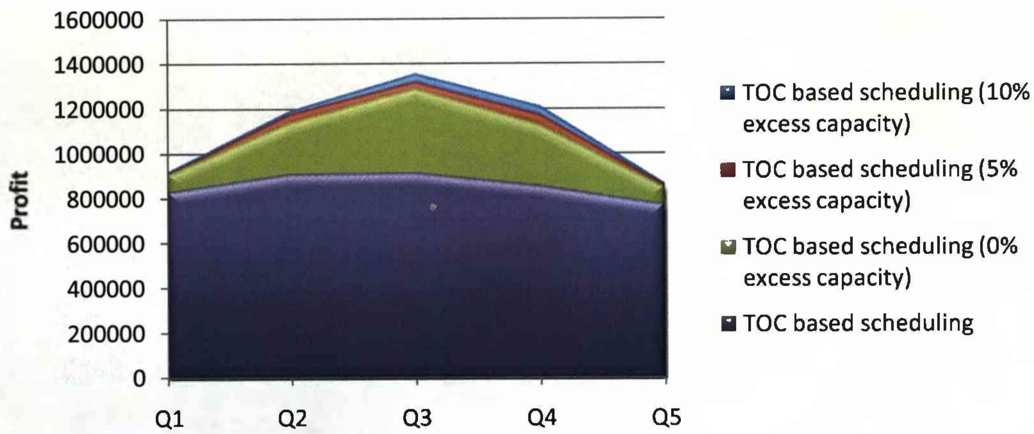


Figure A.3: Profit changes with scheduling rules and machine capacity.

Table A.4: Profit changes by increasing machine capacity under two scheduling rules.

Total profit (£)	Q1	Q2	Q3	Q4	Q5
TOC based scheduling	828405	907595	912075	853420	769385
TOC based scheduling (0% excess capacity)	905715	1138715	1288615	1126495	847265
TOC based scheduling (5% excess capacity)	908785	1177290	1318805	1172910	856010
TOC based scheduling (10% excess capacity)	917815	1190440	1355335	1209955	864920

Shortest processing time	801555	878025	880595	834095	775240
Shortest processing time(0% excess capacity)	934135	1106820	1280195	1185285	849410
Shortest processing time(5% excess capacity)	934135	1127890	1303450	1219540	858815
Shortest processing time(10% excess capacity)	937950	1147240	1334740	1259130	865580



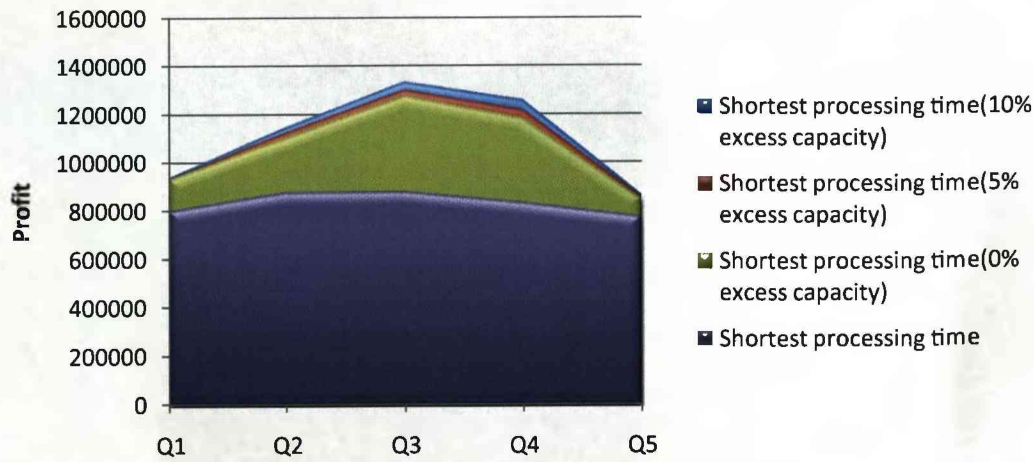
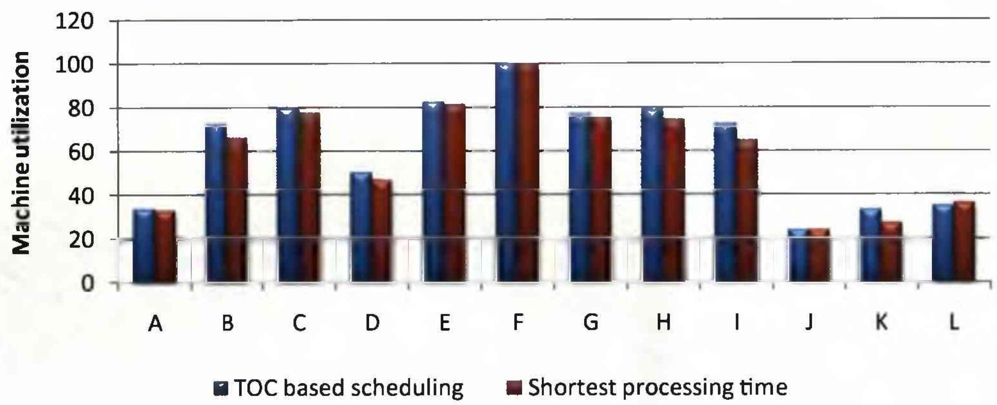


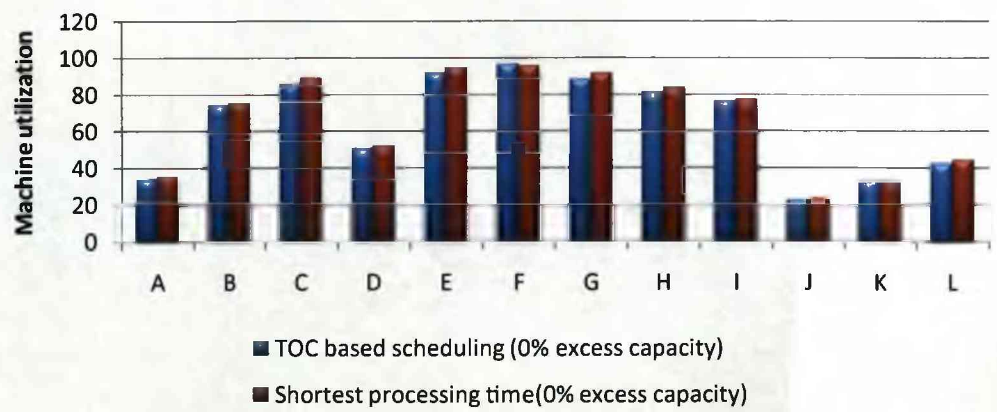
Figure A.4: Profit changes by increasing capacity under two scheduling rules.

Table A.5: Machine utilisation at quarter one.

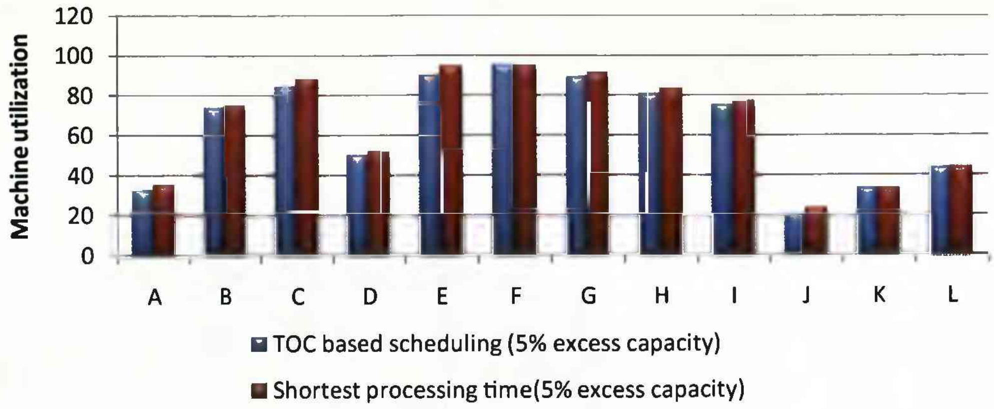
	TOC based scheduling				Shortest Processing Time			
Capacity	*original*	*100%*	*105%*	*110%*	*original*	*100%*	*105%*	*110%*
A	33.37	33.75	32.6	32.87	32.71	35.23	35.23	35.94
B	72.03	74.04	73.49	73.38	66.13	75.15	75.15	76.66
C	79.85	85.3	84.19	84.67	77.14	88.57	88.57	90.45
D	49.91	50.47	49.89	50.52	46.69	51.78	51.78	52.8
E	82.34	91.71	90.04	87.44	81.13	94.63	94.63	92.62
F	100	96.39	95.15	92.35	100	95.41	95.41	94.09
G	76.88	88.05	89.18	88.33	75.45	91.23	91.23	91.49
H	78.97	81.04	80.74	82.69	74.43	83.64	83.64	85.34
I	72.08	76.3	75.11	72.72	64.54	77.08	77.08	78.64
J	23.75	22.44	20.7	20.48	23.75	23.75	23.75	24.2
K	33.33	33.33	33.33	33.33	26.78	33.33	33.33	33.96
L	34.75	42.67	43.94	44.42	36.29	44.11	44.11	45.25



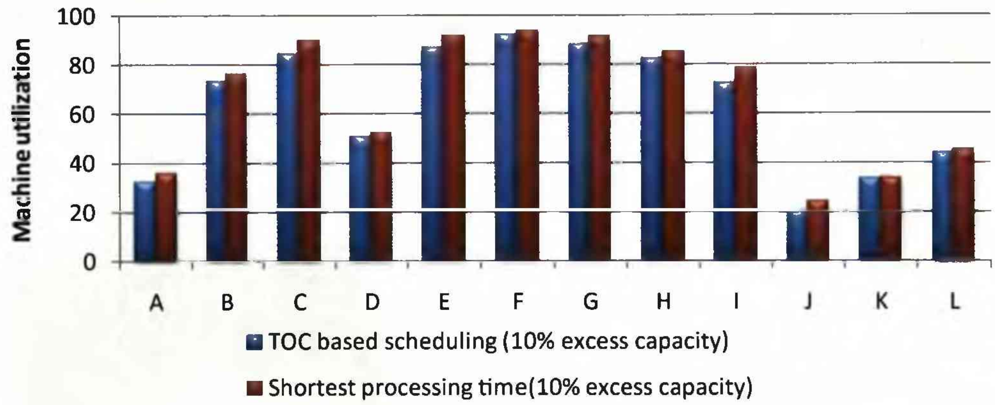
(a)



(b)



(c)

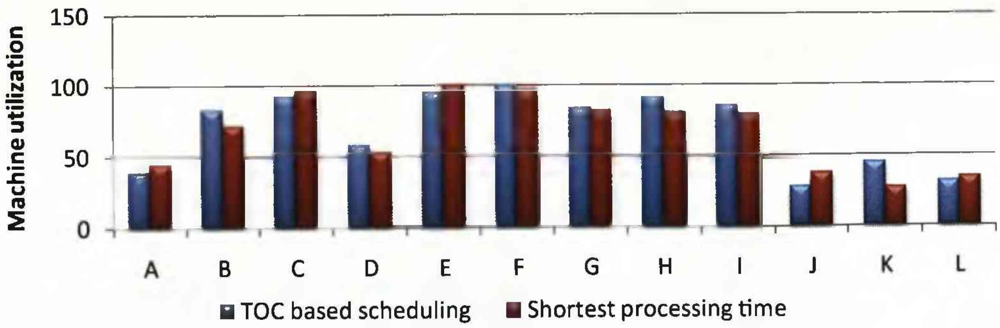


(d)

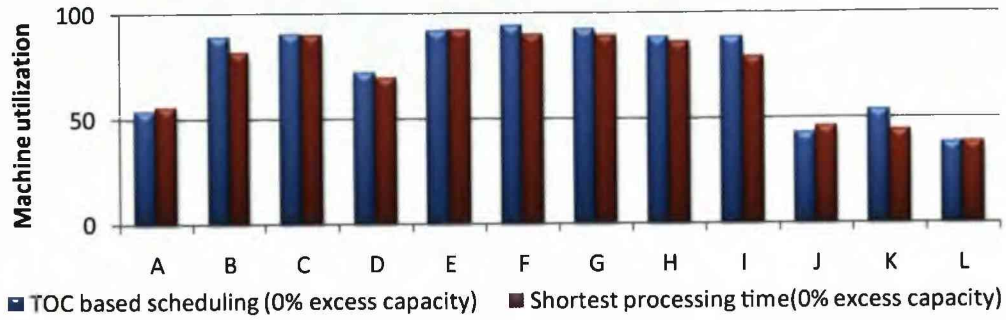
Figure A.5: Machine utilization at Q1.

Table A.6: Machine utilization at Q2.

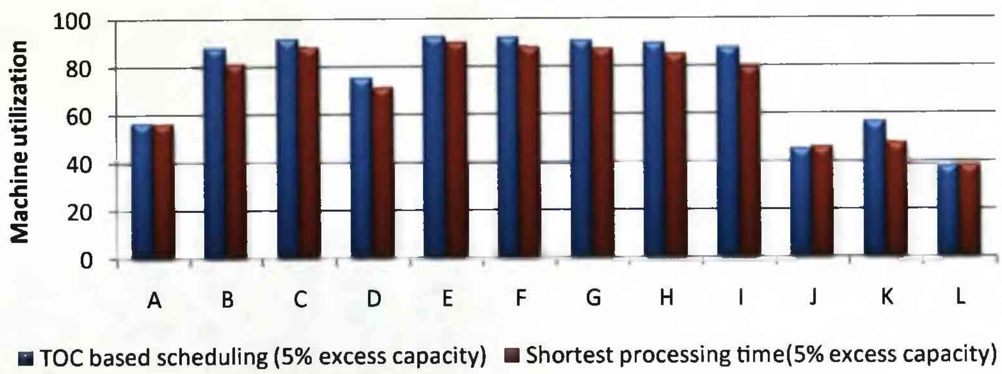
	TOC based scheduling				Shortest Processing Time			
Capacity	*original*	*100%*	*105%*	*110%*	*original*	*100%*	*105%*	*110%*
A	39.04	53.68	56.35	55.77	44.01	55.51	55.98	56.41
B	82.85	88.55	88.02	85.72	72.02	81.82	81.02	80.09
C	92.76	90.61	91.17	88.05	96.09	89.95	88.2	86.49
D	58.27	72.07	75.44	76.89	52.79	69.65	71.37	72.96
E	94.94	91.57	92.16	89.36	100	91.95	89.97	88.03
F	100	94.28	91.93	89.76	96.96	89.89	87.91	86
G	83.92	92.03	90.95	88.3	82.28	89.3	87.15	85.08
H	92.11	88.65	89.47	88.94	81.06	86.35	85.02	83.66
I	85.74	88.63	87.89	83.57	79.55	79.25	79.33	79.2
J	28.19	42.74	45.02	43.2	37.62	46.08	46.08	46.08
K	44.74	53.55	56.17	58.87	27.34	44.17	47.31	50.22
L	31.62	37.92	37.92	37.92	34.6	37.92	37.92	37.92



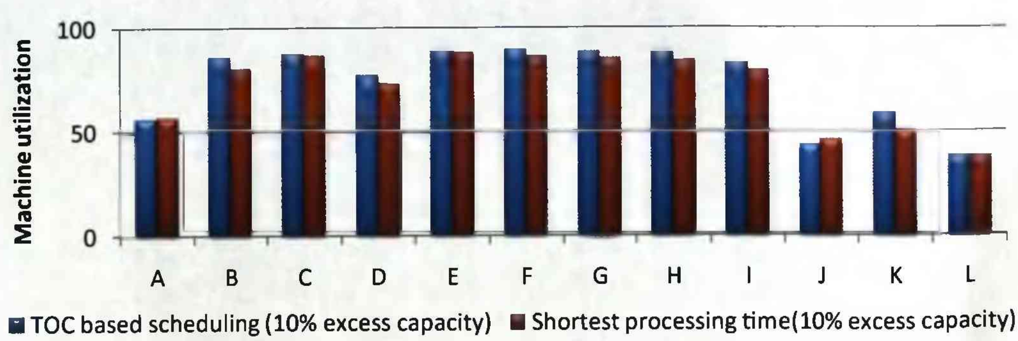
(a)



(b)



(c)

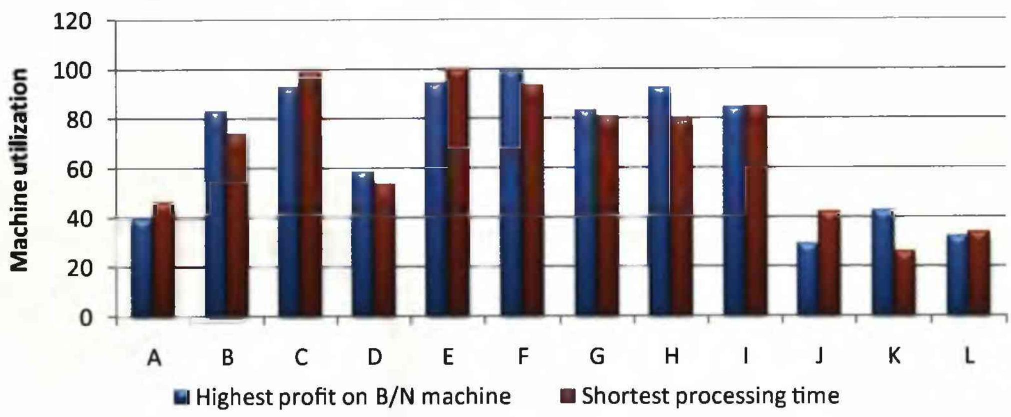


(d)

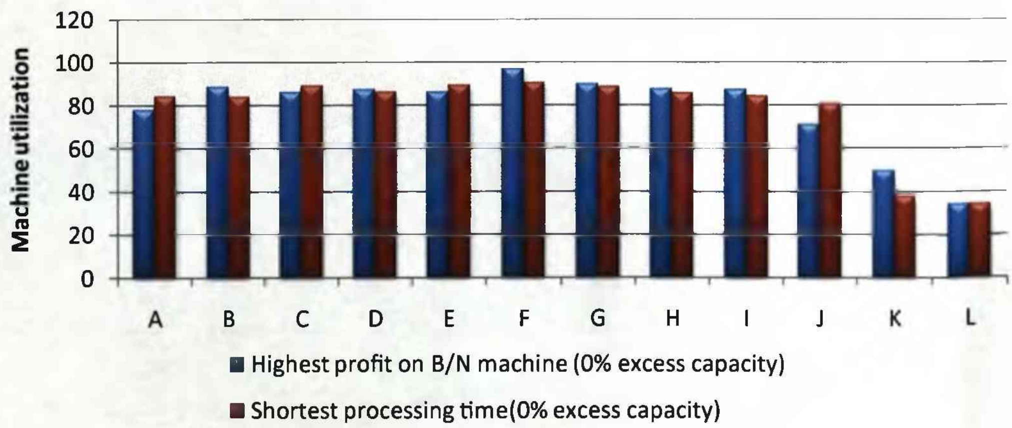
Figure A. 6: Machine utilization at Q2.

Table A.7: Machine utilization at Q3.

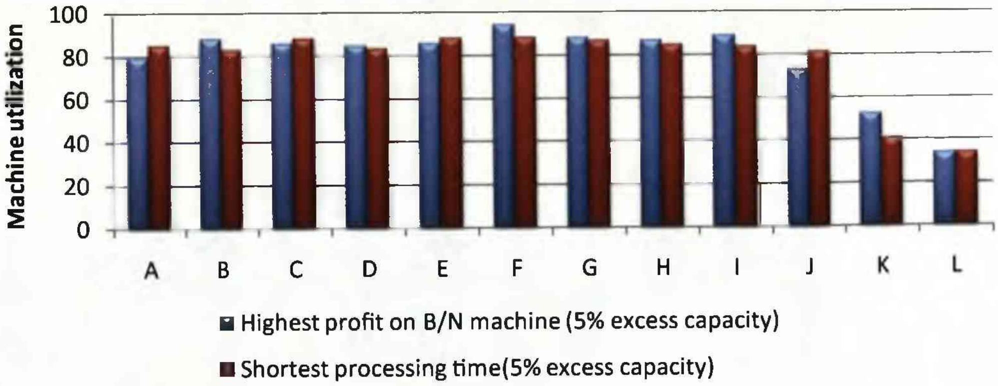
	TOC based scheduling				Shortest Processing Time			
Capacity	*original*	*100%*	*105%*	*110%*	*original*	*100%*	*105%*	*110%*
A	39.86	77.67	79.62	79.95	46.45	84.27	85.13	84.19
B	83.1	88.65	88.27	87.61	74.12	83.89	83.07	82.69
C	93.01	85.78	85.83	86.33	98.34	89.15	88.15	87.82
D	58.7	87.17	84.78	83.81	53.64	86.08	83.47	82.09
E	94.43	85.77	85.49	86.05	99.98	89.41	88.3	87.92
F	100	96.73	93.97	91.18	93.27	90.43	88.05	85.93
G	83.42	89.71	88.25	87.3	80.48	88.57	86.8	85.6
H	92.4	87.85	86.71	87.17	79.85	85.59	84.81	84.78
I	84.7	87.38	89.09	88.37	84.98	84.18	83.87	83.74
J	29.4	70.83	72.91	75.41	41.84	80.95	81.38	82.52
K	42.78	49.47	52.2	54.37	25.9	37.83	40.74	43.82
L	32.31	33.71	33.71	33.71	33.71	33.71	33.71	33.71



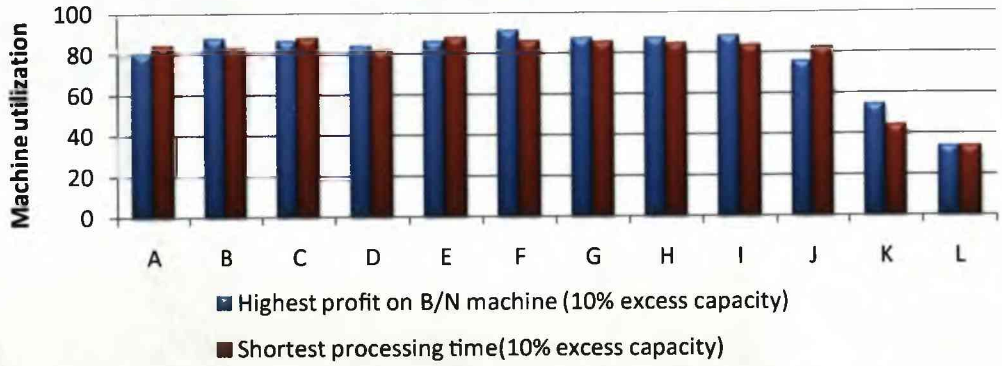
(a)



(b)



(c)

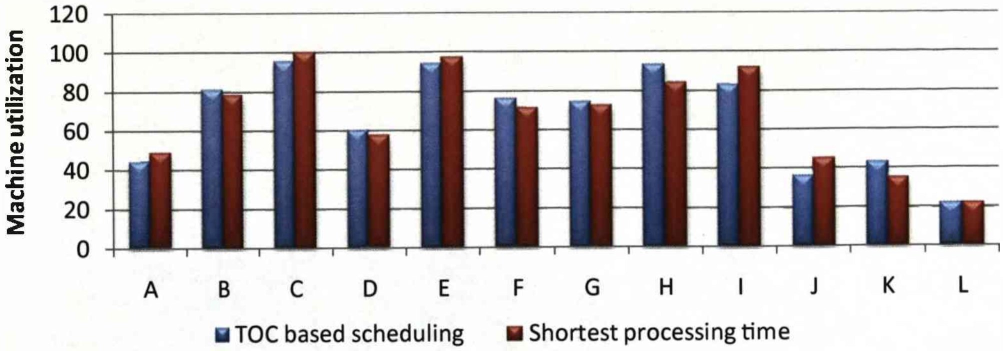


(d)

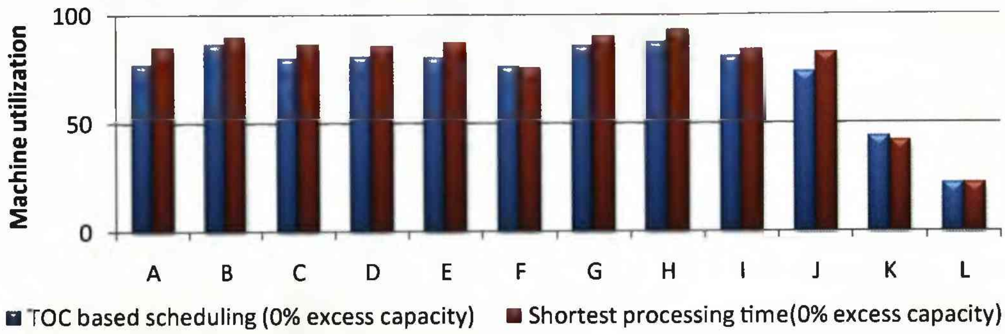
Figure A. 7: Machine utilization at Q3.

Table A. 8: Machine utilization at Q4.

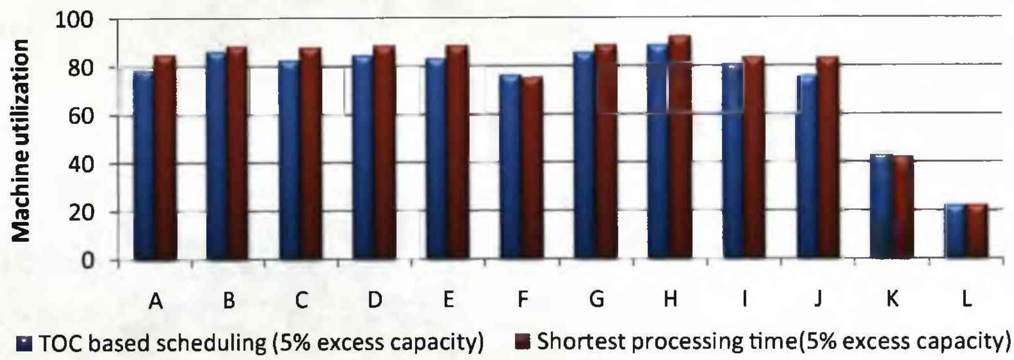
	TOC based scheduling				Shortest Processing Time			
Capacity	*original*	*100%*	*105%*	*110%*	*original*	*100%*	*105%*	*110%*
A	44.21	77.09	78.49	78.45	48.54	85.02	84.71	84.52
B	81.13	86.85	85.93	84.63	78.45	89.64	88.16	86.99
C	95.42	79.97	82.38	83.81	99.86	86.53	87.62	89.1
D	60.21	80.63	84.48	83.32	57.21	85.53	88.2	87.11
E	94.46	80.25	82.83	84.34	97.22	87.15	88.24	89.8
F	76.03	76.03	76.03	76.03	71.14	75.07	75.13	75.26
G	74.59	85.98	85.5	84.43	72.44	90.17	88.63	87.62
H	93.57	87.72	88.65	88.51	83.8	92.94	92.12	92.17
I	83.09	80.87	80.68	80.36	91.91	83.53	83.48	83.22
J	36.21	73.51	75.59	76.53	45	82.37	82.78	83.53
K	43.33	43.33	43.33	43.33	35	41.56	41.67	41.86
L	22.13	22.13	22.13	22.13	22.13	22.13	22.13	22.13



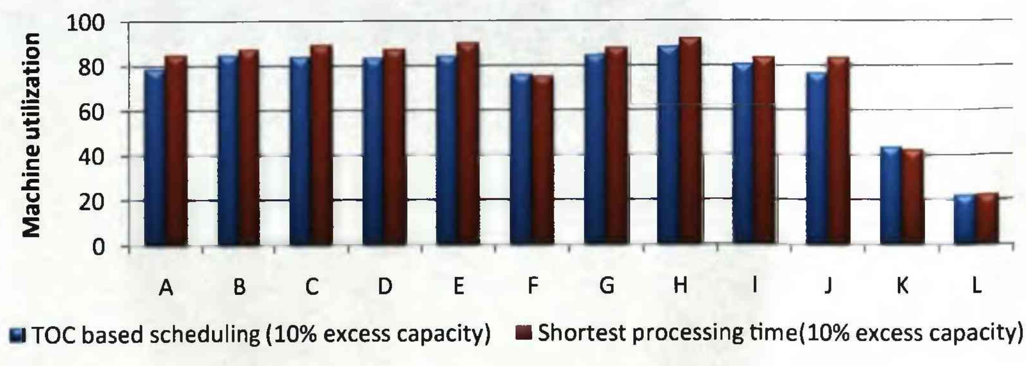
(a)



(b)



(c)

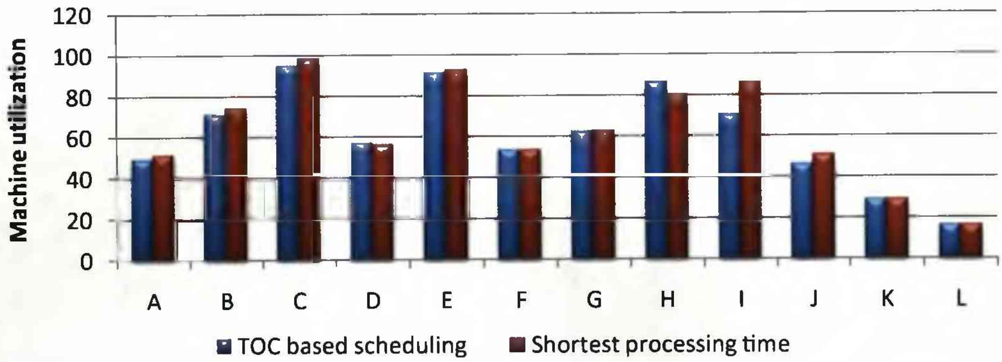


(d)

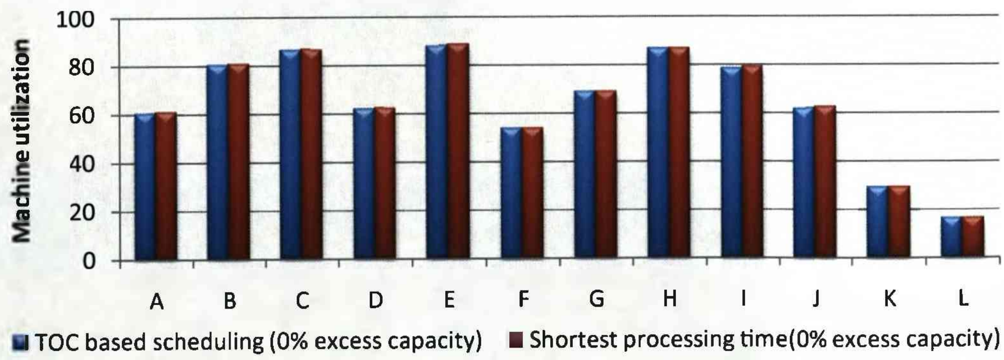
Figure A.8: Machine utilization at Q4.

Table A.9: Machine utilization at Q5.

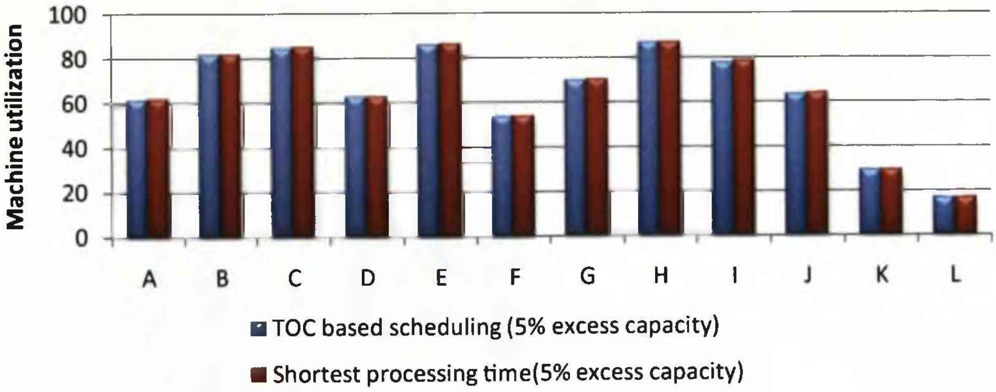
	TOC based scheduling				Shortest Processing Time			
Capacity	*original*	*100%*	*105%*	*110%*	*original*	*100%*	*105%*	*110%*
A	49.57	60.15	61.34	62.54	51.51	60.43	61.73	62.66
B	71.38	80.46	81.48	82.52	74.47	80.7	81.79	82.6
C	94.74	86.16	84.49	82.96	98.53	86.54	84.98	83.11
D	57.36	61.9	62.41	62.92	56.53	62.01	62.57	62.96
E	91.39	88.02	85.87	83.9	92.82	88.35	86.31	84.02
F	53.61	53.61	53.61	53.61	53.61	53.61	53.61	53.61
G	62.85	68.9	69.58	70.27	63.22	69.07	69.8	70.33
H	86.78	86.78	86.78	86.78	80.72	86.78	86.78	86.78
I	70.93	78.53	77.84	77.29	86.54	79.12	78.62	77.51
J	46.41	61.54	63.24	64.97	50.87	61.96	63.78	65.13
K	29.17	29.17	29.17	29.17	29.17	29.17	29.17	29.17
L	16.58	16.58	16.58	16.58	16.58	16.58	16.58	16.58



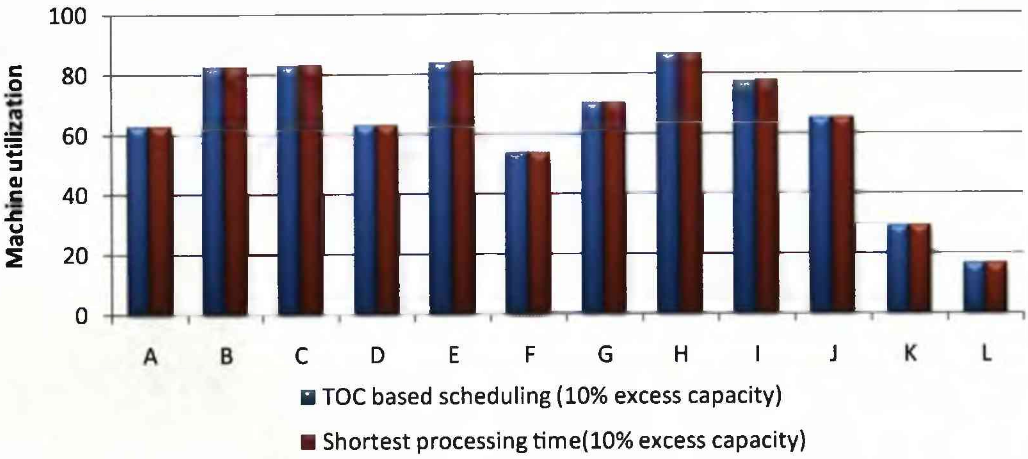
(a)



(b)



(c)



(d)

Figure A.9: Machine utilization at Q5.

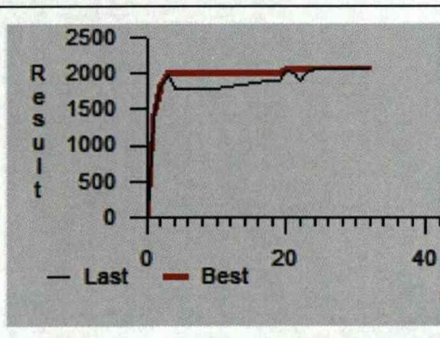
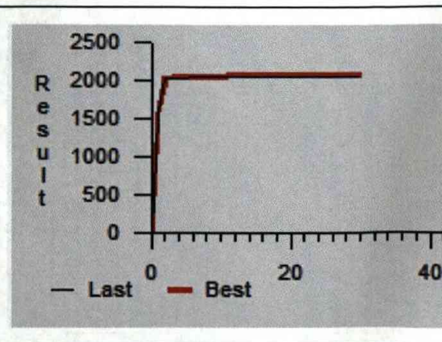
Appendix B
Optimisation on WIP level

Further optimisation experiments were carried out under system configurations in order to cover more scenarios and support the argument that system performs the best when *wip* value is 3. Using the difference between the market demand and existing machine capacity calculates the suggested improvements on each bottleneck machine. The following thirty experiments are under the suggested capacity reconfiguration, 5% excess the suggested capacity and 10% excess capacity. Same capacity reconfiguration was used in scheduling rules experiments as well. Table B.1 is suggested improvement for all twelve machines.

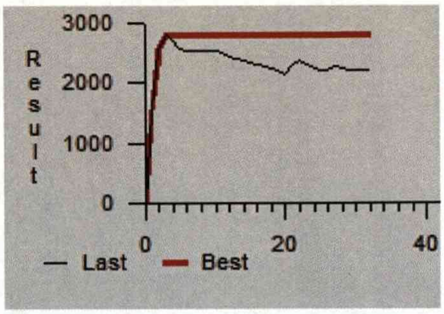
Table B.1: Suggested machine improvement to meet the market demand.

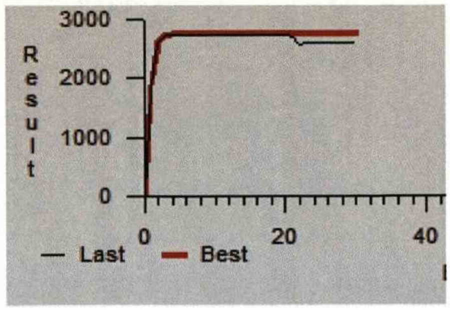
	Q1	Q2	Q3	Q4	Q5
A				0.50%	
B		14.92%	26.54%	19.96%	
C		36.79%	84.21%	88.42%	30.58%
D			1.54%		
E		40.29%	86.63%	83.33%	21.25%
F	23.88%	27.58%	10.71%		
G		16.13%	27.96%	11.50%	
H		26.02%	54.33%	32.31%	
I		30.08%	41.00%	59.58%	24.67%
J				5.25%	
K					
L					

Table B.2: Optimisation results on different WIP level after machine reconfiguration.

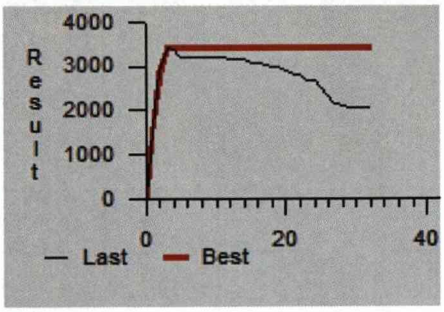
	Highest profit on B/N machine (0% excess capacity)	**Shortest processing time (0% excess capacity)**
Quarter 1		

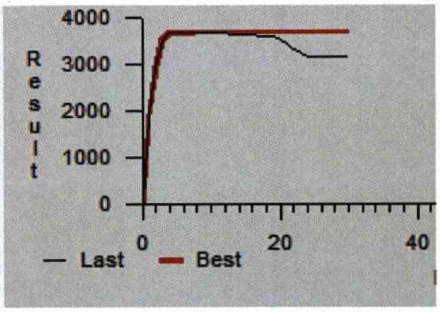
Quarter 2



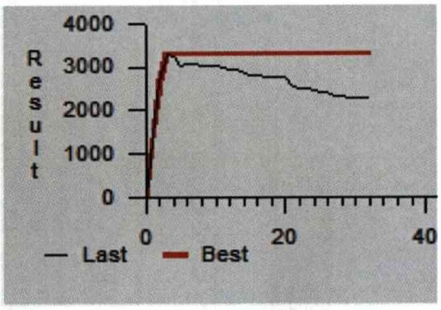


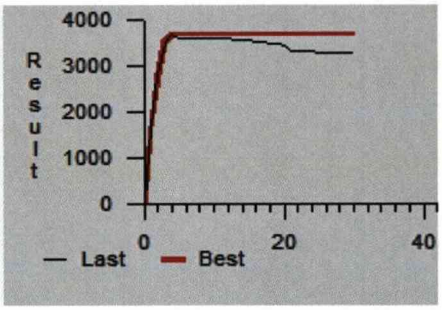
Quarter 3



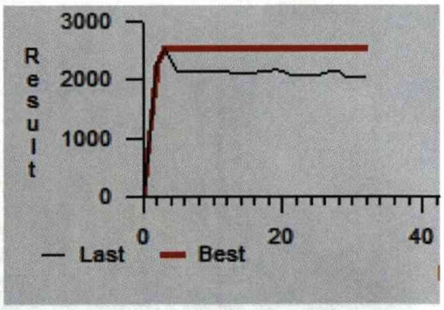


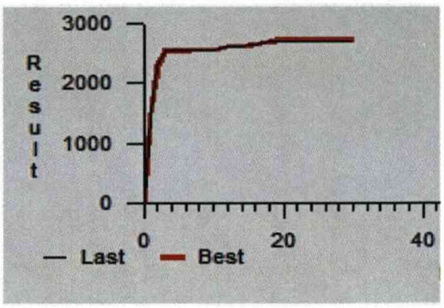
Quarter 4





Quarter 5





Further optimisation experiments have been done to test WIP effects. The bottleneck machines capacities are future improved in the following experiments. Machine capacities now have 5% excess and 10% excess.

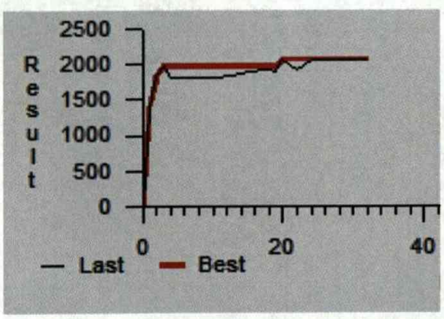
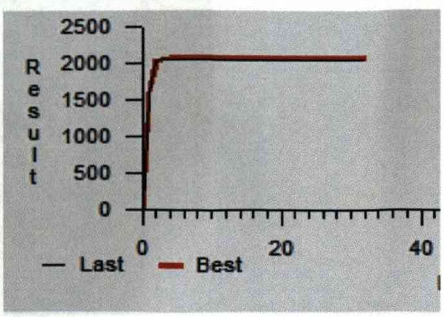
Table B.3: 5% excess suggested machine improvement to meet the market demand.

	Q1	Q2	Q3	Q4	Q5
A				5.50%	
B		19.92%	31.54%	24.96%	
C		41.79%	89.21%	93.42%	35.58%
D			6.54%	0.08%	
E	1.46%	45.29%	91.63%	88.33%	26.25%
F	28.88%	32.58%	15.71%		
G		21.13%	32.96%	16.50%	
H		31.02%	59.33%	37.31%	
I		35.08%	46.00%	64.58%	29.67%
J				10.25%	
K					
L					

Table B.4: 10% excess suggested machine improvement to meet the market demand.

	Q1	Q2	Q3	Q4	Q5
A			2.54%	10.50%	
B		24.92%	36.54%	29.96%	
C		46.79%	94.21%	98.42%	40.58%
D			11.54%	5.08%	
E	6.46%	50.29%	96.63%	93.33%	31.25%
F	33.88%	37.58%	20.71%		
G*2	1.69%	26.13%	37.96%	21.50%	
H*2		36.02%	64.33%	42.31%	
I		40.08%	51.00%	69.58%	34.67%
J				15.25%	
K					
L					

Table B. 5: Optimisation results on different WIP level after machine reconfiguration

	Highest profit on B/N machine (5% excess capacity)	**Shortest processing time (5% excess capacity)**
Quarter 1		

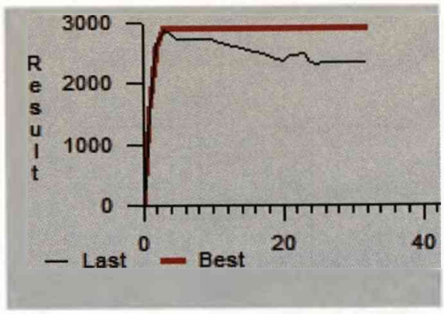
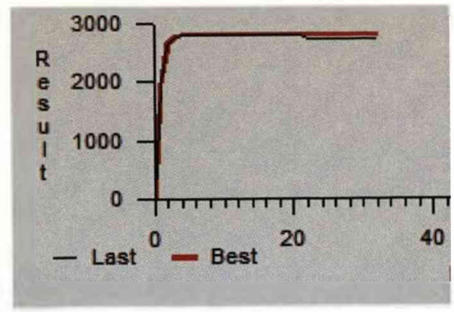
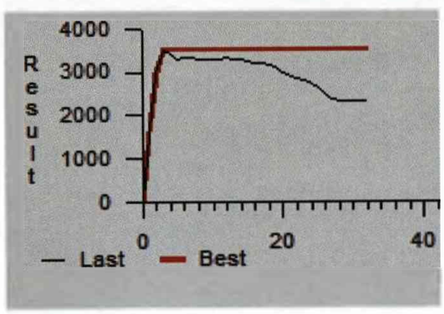
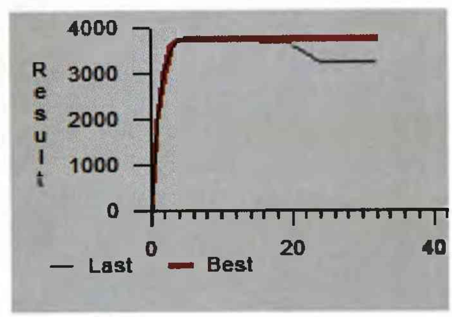
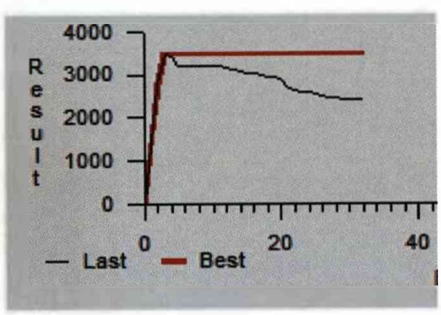
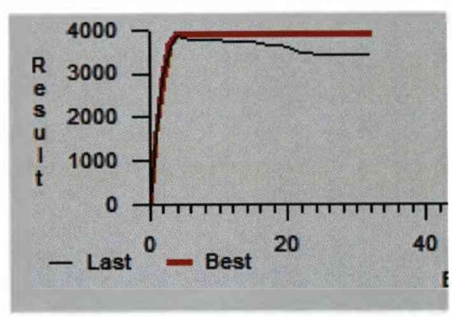
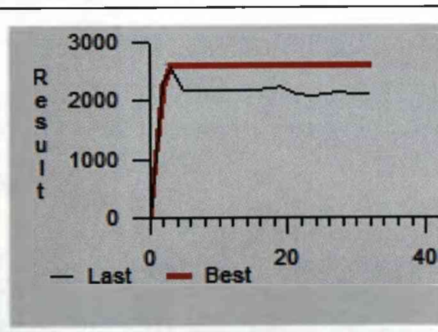
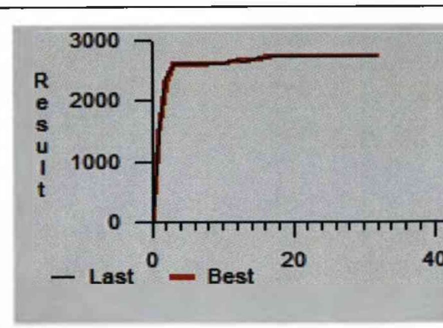
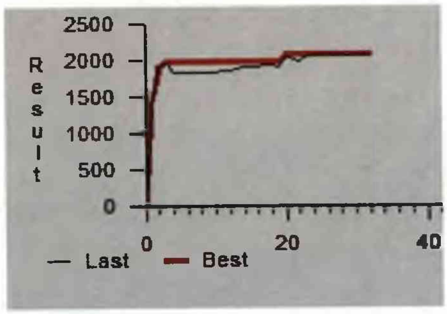
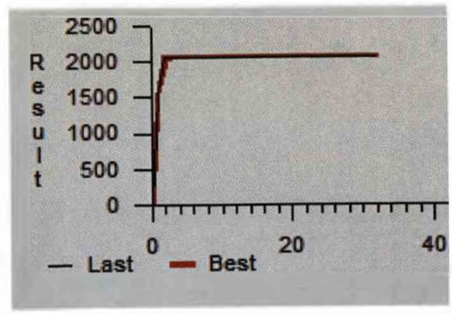
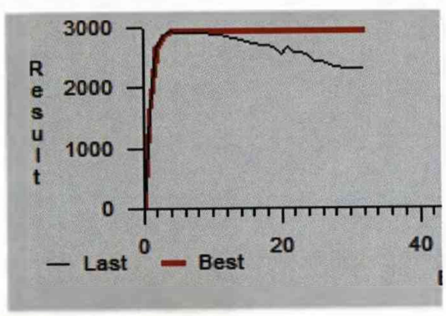
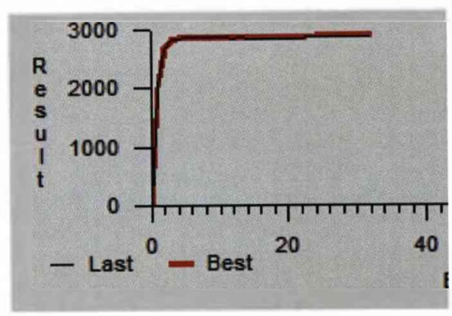
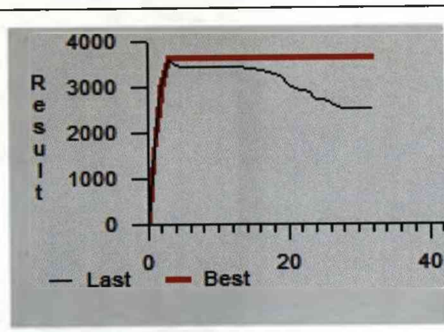
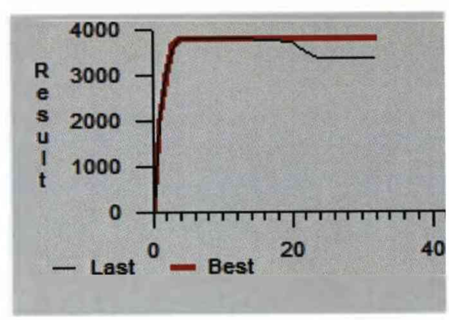
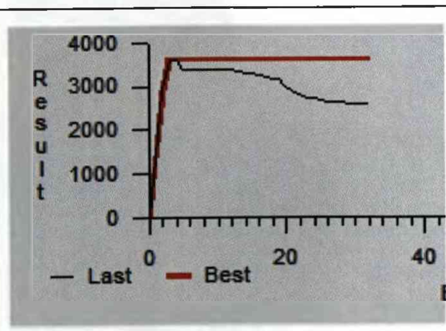
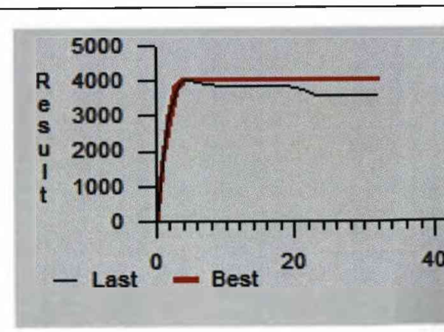
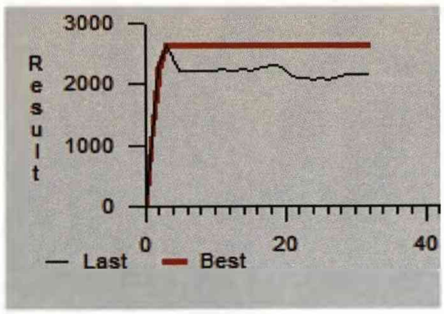
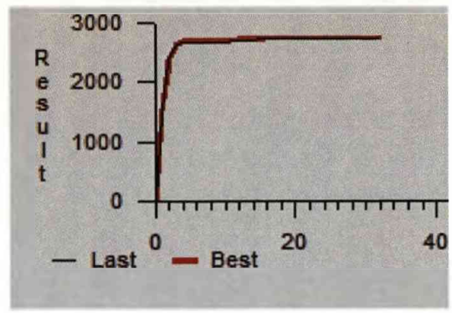
Quarter 2		
Quarter 3		
Quarter 4		
Quarter 5		

Table B.6: Optimisation results on different WIP level after machine reconfiguration.

<table>
<tr><th></th><th>Highest profit on B/N machine
(10% excess capacity)</th><th>Shortest processing time
(10% excess capacity)</th></tr>
<tr><td>Quarter
1</td><td>
</td><td>
</td></tr>
<tr><td>Quarter
2</td><td>
</td><td>
</td></tr>
<tr><td>Quarter
3</td><td>
</td><td>
</td></tr>
<tr><td>Quarter
4</td><td>
</td><td>
</td></tr>
</table>

Quarter 5		

From the results above it can be concluded that wip equal 3 is the best WIP level choice for the particular model.

Appendix C
Reconfiguration cost models

The following charts are generated for the nine machine reconfiguration cost models; three additional situations are added under linear, non-linear concave and non-linear convex categories. During the reconfiguration of manufacturing system, it is often found that bottleneck machines are high cost machine as that's why they are still not been rescaled. Three additional cost models for high cost bottleneck machines and low cost non-bottleneck machines are listed in the appendix.

Parameter k is generated randomly so that the machine costs are within the range of low cost or high cost machines in UK market. There are two charts under each cost model to demonstrate the randomness of the parameter k.

1. ***Linear low cost machine model***

   $C(x) = kx$

   Parameter k is randomly generated between 0.2 to 0.3. (Machine cost is within the range of £20,000 to £30,000)

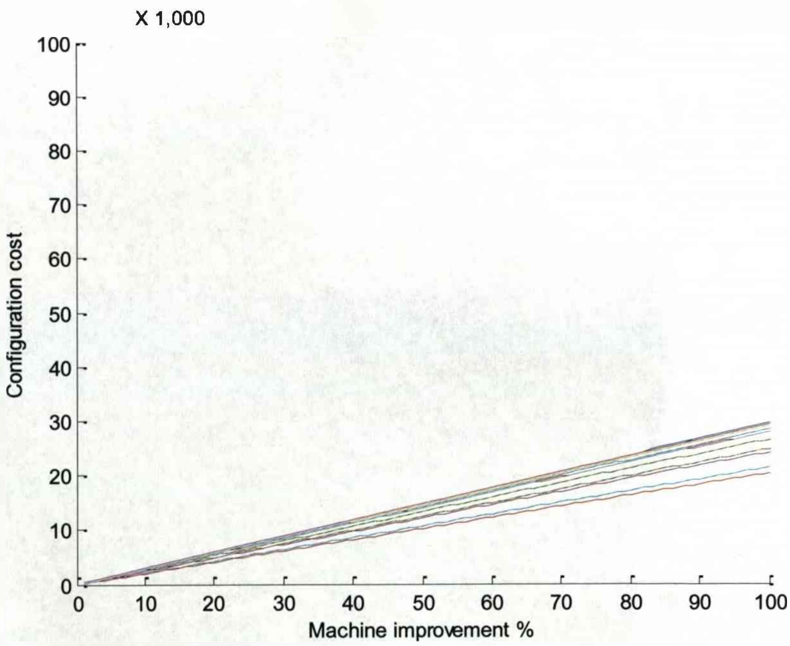


Figure C.1: (Example 1) Random linear low cost machine model.

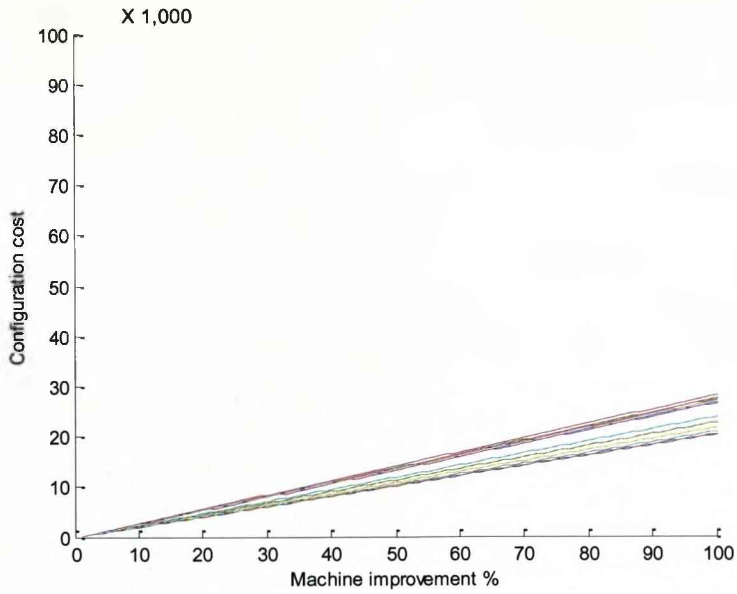


Figure C.2: (Example 2) Random linear low cost machine model.

2. ***Linear high cost machine model***

$C(x) = kx$

Parameter k is randomly generated between 0.5 to 0.75. (Machine cost is within the range of £50,000 to £75,000)

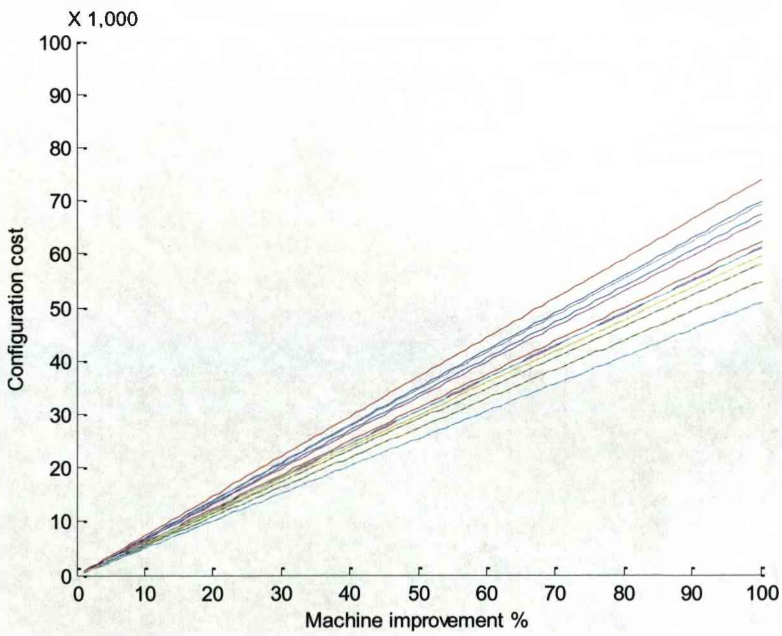


Figure C.3: (Example 1) Random linear high cost machine model.

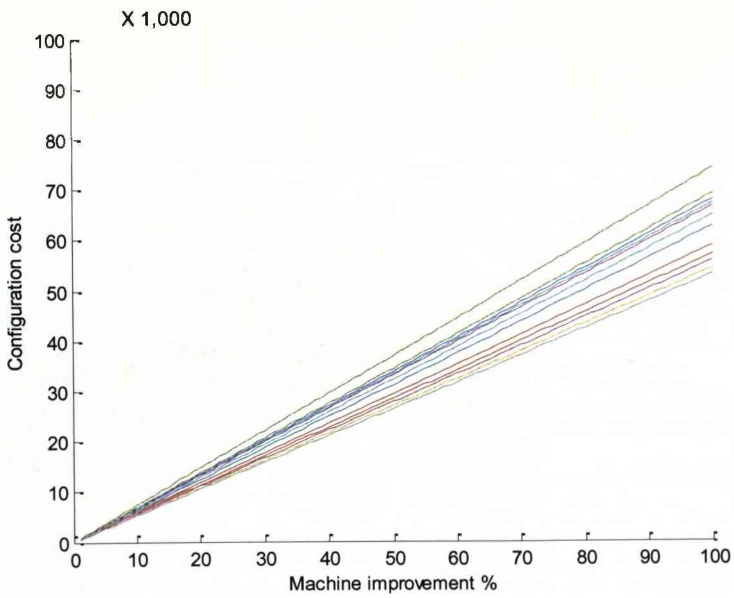


Figure C.4: (Example 2) Random linear high cost machine model.

3. ***Linear vary cost machine model***

$C(x) = kx$

Parameter k is randomly generated between 0.2 to 0.75. (Machine cost is within the range of £20,000 to £75,000)

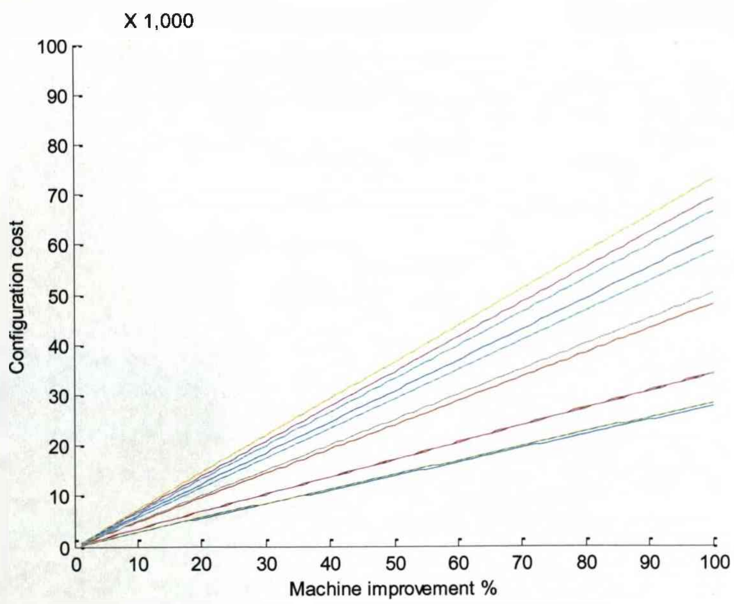


Figure C.5: (Example 1) Random linear vary cost machine model.

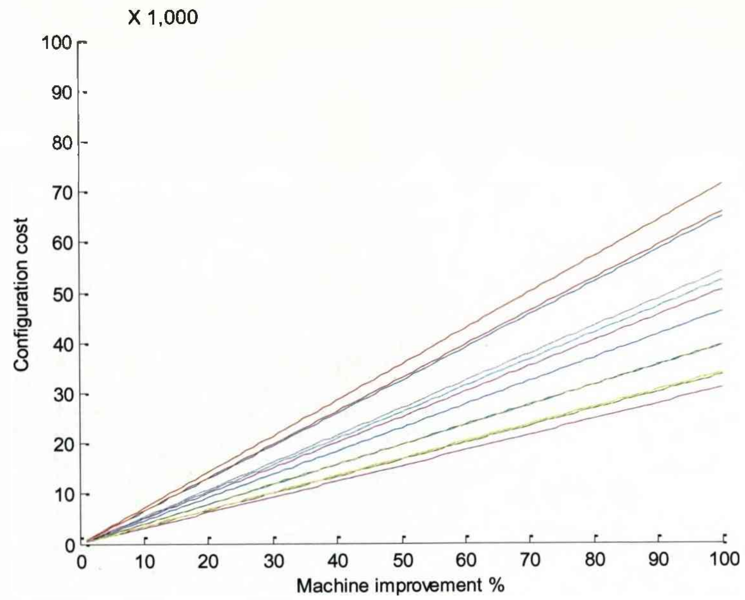


Figure C.6: (Example 2) Random linear vary cost machine model.

4. ***Linear relationship where bottleneck machines are high cost machines:***

   $C(x) = kx$

   Parameter k is randomly generated between 0.5 and 0.75 for bottleneck machines and 0.2 and 0.30 for non-bottleneck machines.

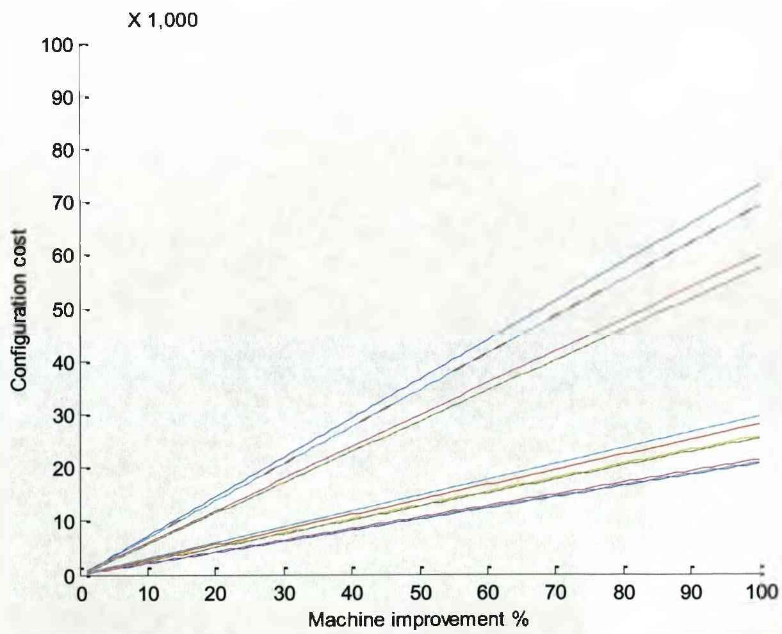


Figure C.7: (Example 1) Random linear relationships (Bottleneck machines are high cost).

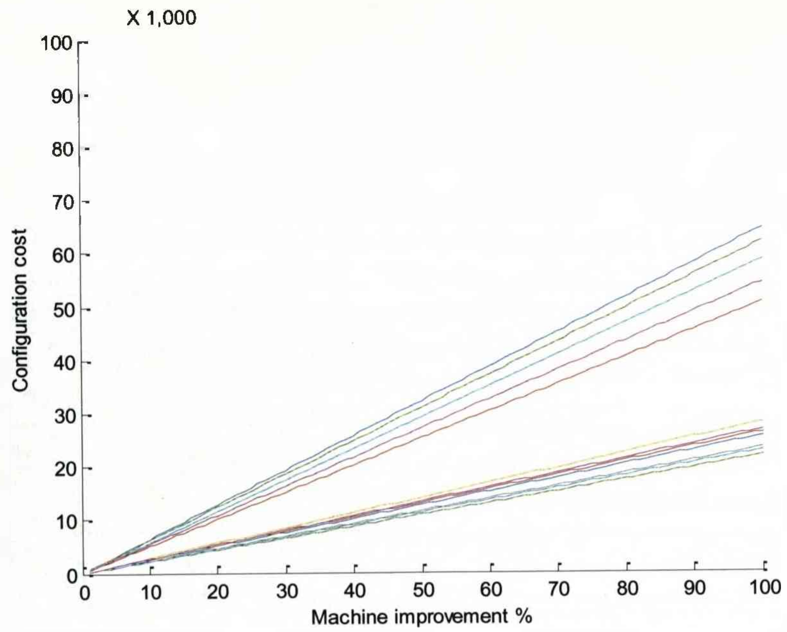


Figure C.8: (Example 2) Random linear relationships (Bottleneck machines are high cost).

5. ***Non-linear concave low cost machine model***

$C(x) = kx^{1/2}$

Parameter k is randomly generated between 2 to 3. (Machine cost is within the range of £20,000 to £30,000)

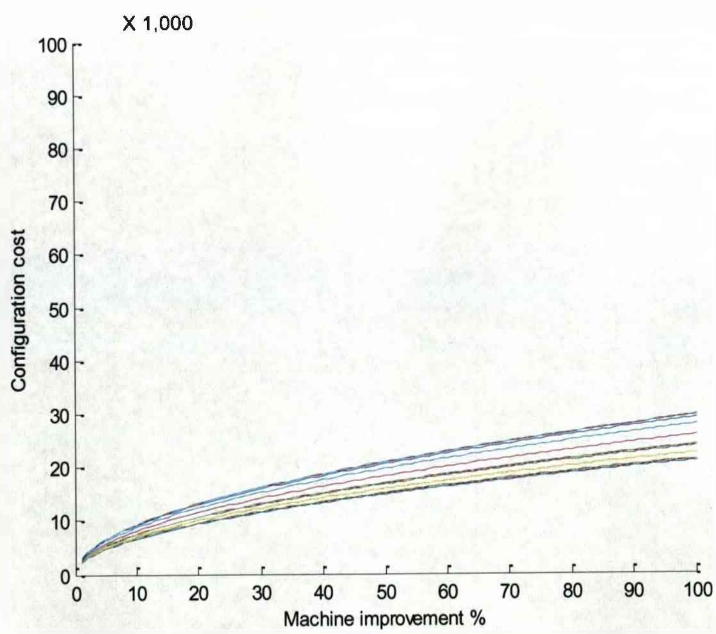


Figure C.9 : (Example 1) Random non-linear concave low cost machine model.

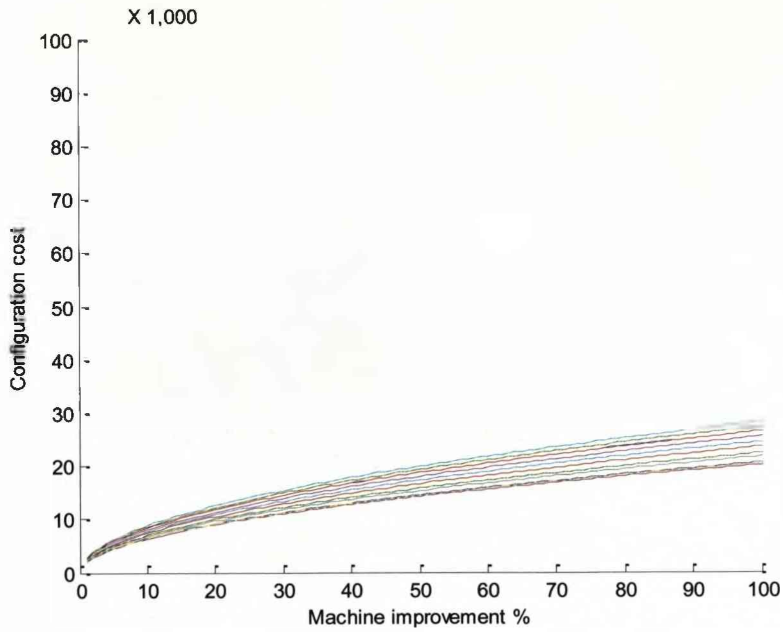


Figure C.10: (Example 2) Random non-linear concave low cost machine model.

6. ***Non-linear concave high cost machine model***

$C(x) = kx^{1/2}$

Parameter k is randomly generated between 5 to 7.5. (Machine cost is within the range of £50,000 to £75,000)

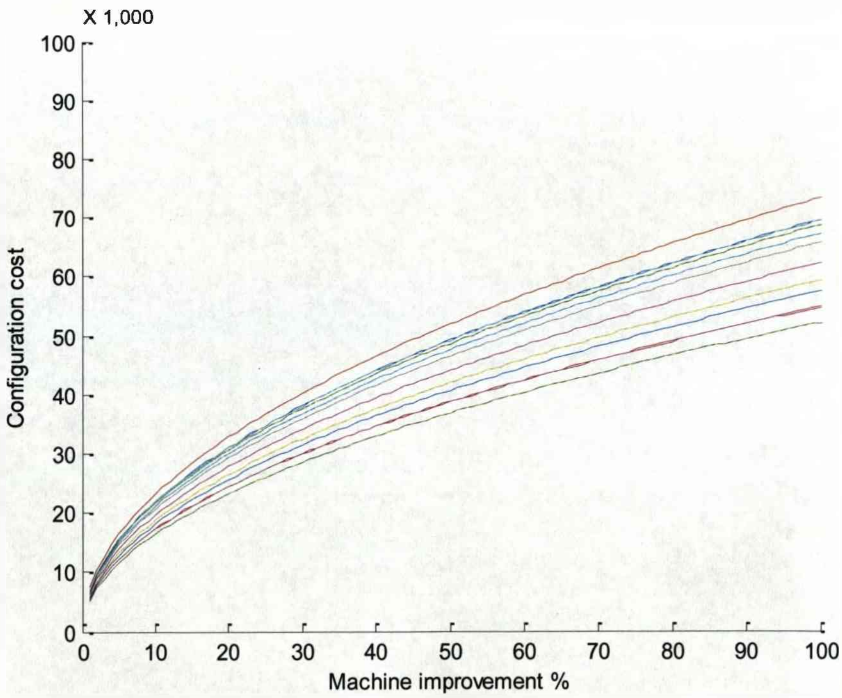


Figure C.11: (Example 1) Random non-linear concave high cost machine model.

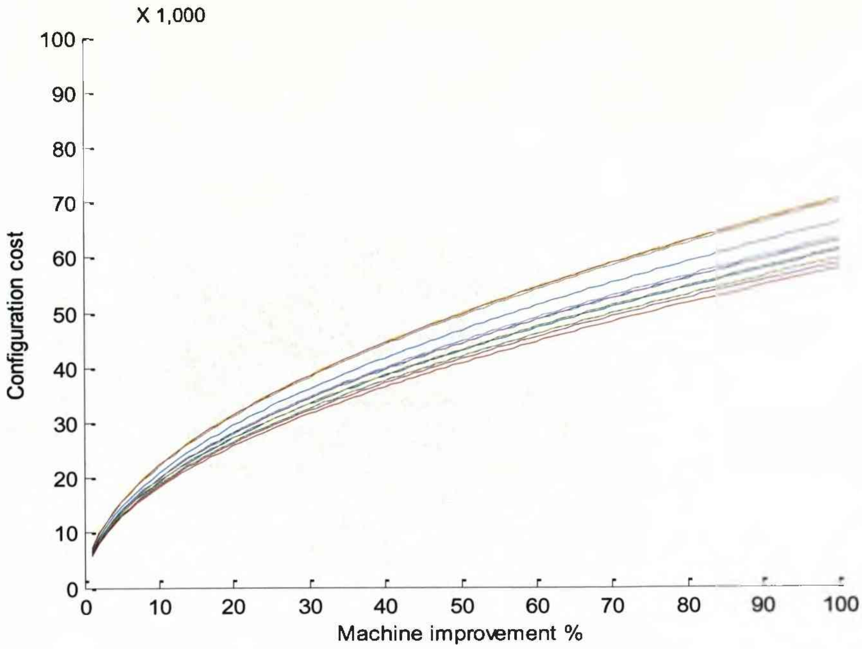


Figure C.12: (Example 2) Random non-linear concave high cost machine model.

7. ***Non-linear concave vary cost machine model***

   $C(x) = kx^{1/2}$

   Parameter k is randomly generated between 2 to 7.5 . (Machine cost is within the range of £20,000 to £75,000)

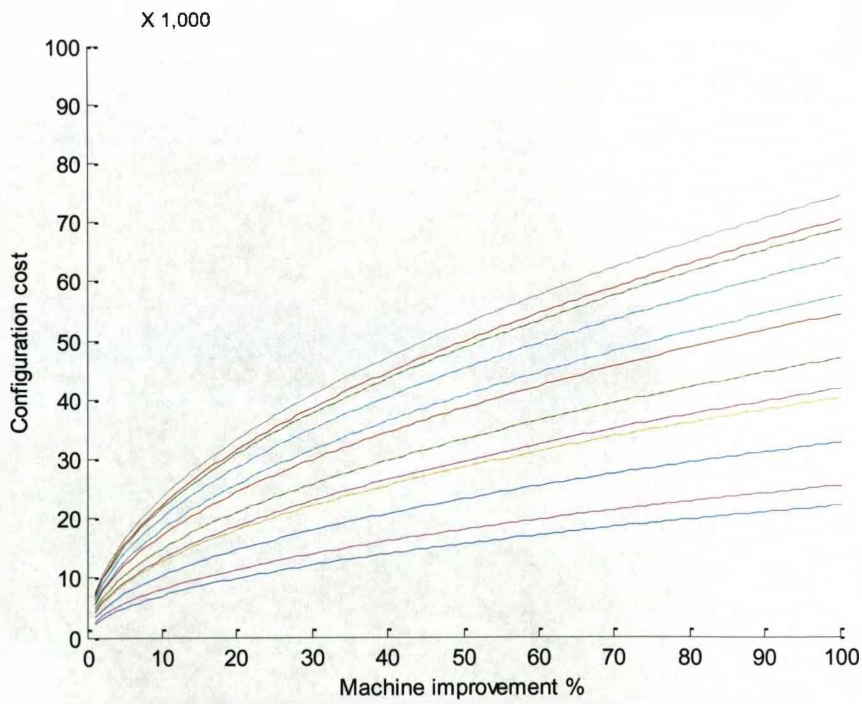


Figure C.13 : (Example 1) Random non-linear concave vary cost machine model.

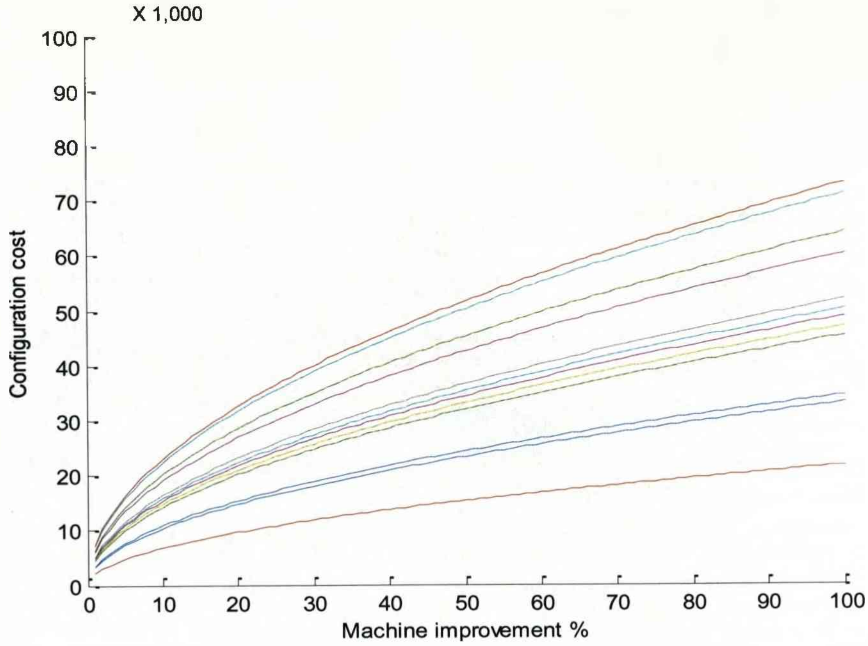


Figure C.14: (Example 2) Random non-linear concave vary cost machine model.

8. ***Non-linear concave model where bottleneck machines are high cost machines***

   $C(x) = kx^{1/2}$

   Parameter k is randomly generated between 5 to 7.5 for bottleneck machines and 2 to 3 for non-bottleneck machines.

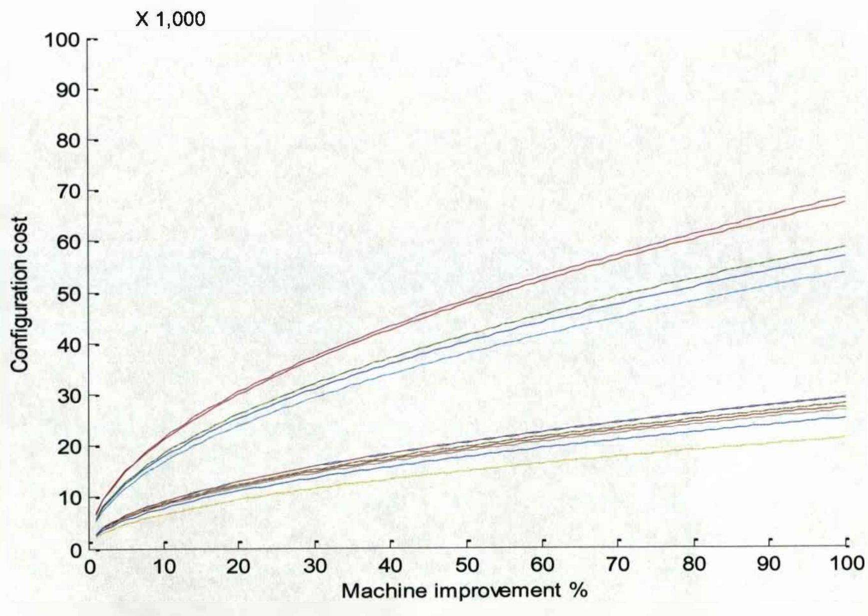


Figure C.15: (Example 1) Random concave non-linear relationships (Bottleneck machines are high cost).

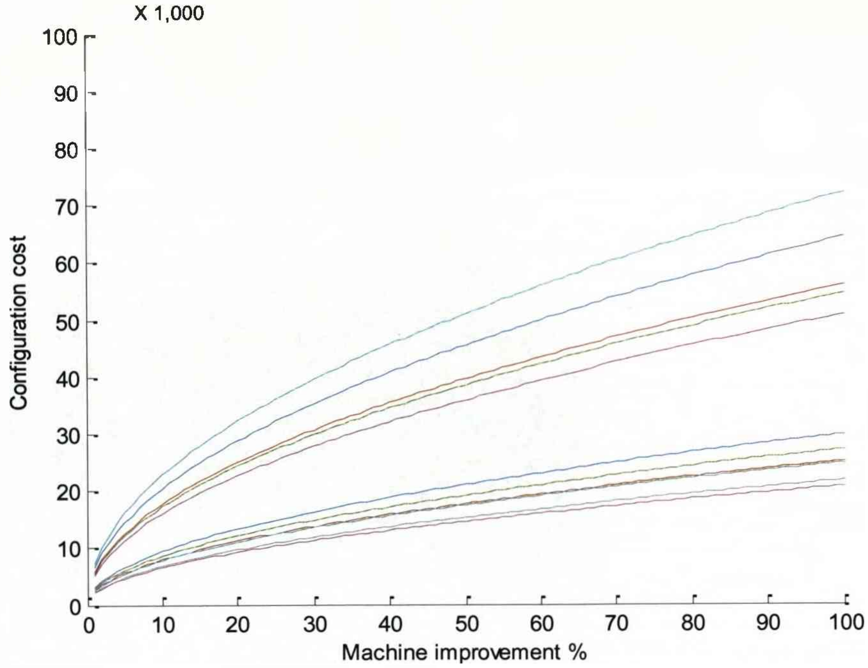


Figure C.16 : (Example 2) Random concave non-linear relationships (Bottleneck machines are high cost).

9. ***Non-linear convex low cost machine model***

$C(x) = kx^2$

Parameter k is randomly generated between 0.002 to 0.003. (Machine cost is within the range of £20,000 to £30,000)

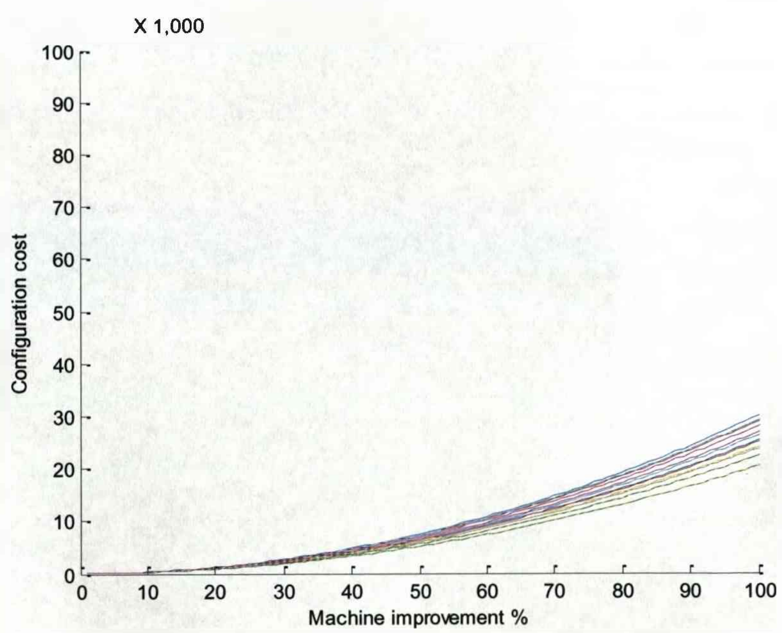


Figure C.17: (Example 1) Random convex non-linear low cost machine model.

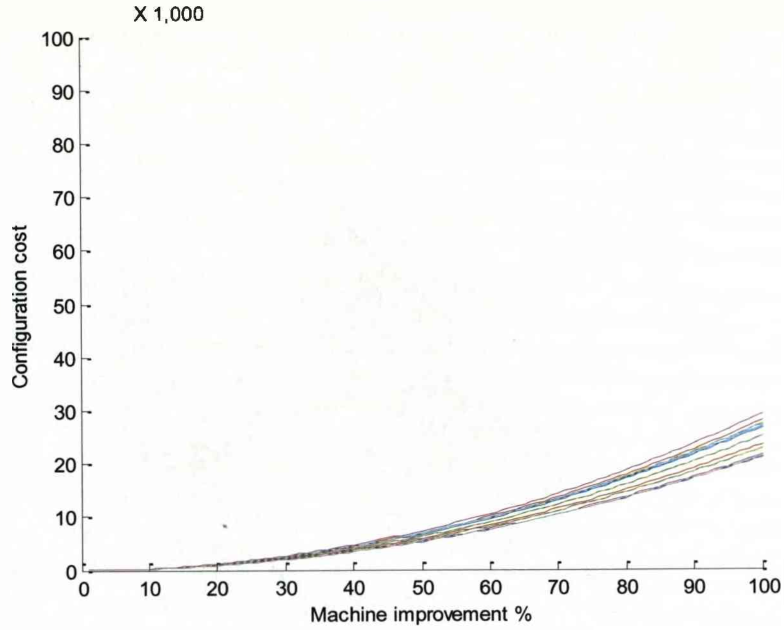


Figure C.18: (Example 2) Random convex non-linear low cost machine model.

10. Non-linear convex high cost machine model

$C(x) = kx^2$

Parameter k is randomly generated between 0.005 to 0.0075. (Machine cost is within the range of £50,000 to £75,000)

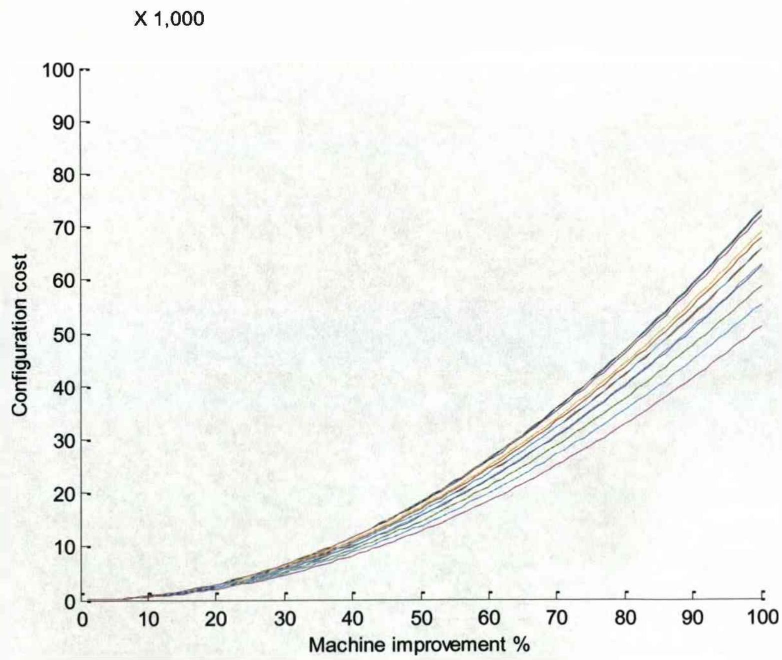


Figure C.19 : (Example 1) Random convex non-linear high cost machine model.

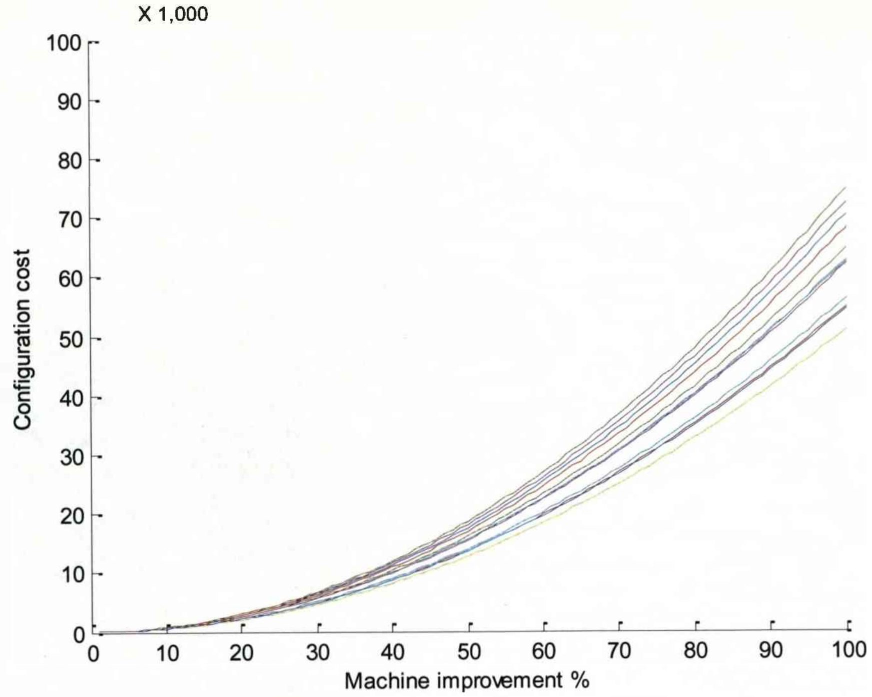


Figure C.20: (Example 2) Random convex non-linear high cost machine model.

11. Non-linear convex vary cost machine model

$C(x) = kx^2$

Parameter k is randomly generated between 0.002 to 0.0075 . (Machine cost is within the range of £20,000 to £75,000)

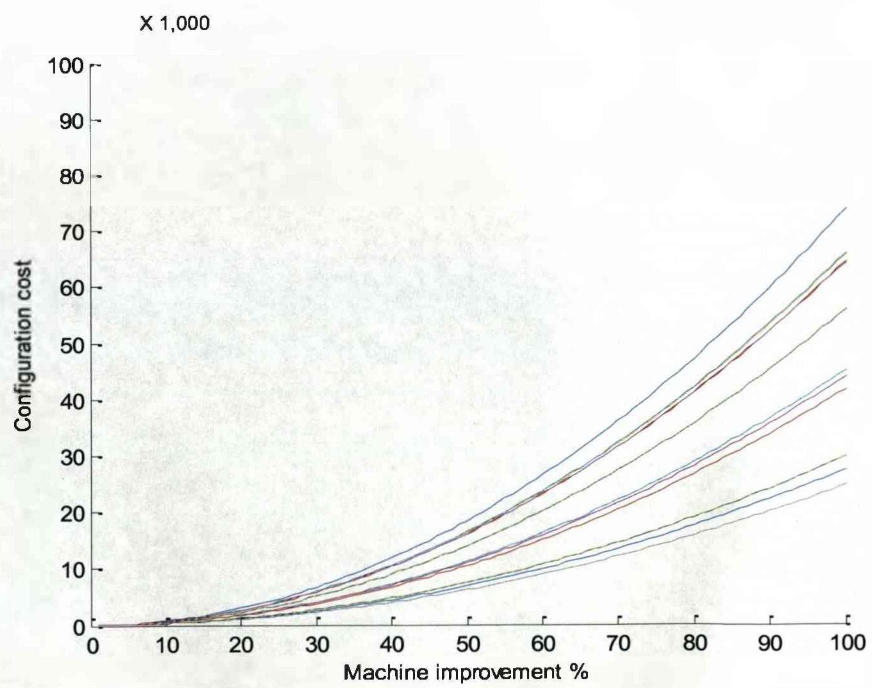


Figure C.21: (Example 1) Random convex non-linear vary cost machine model.

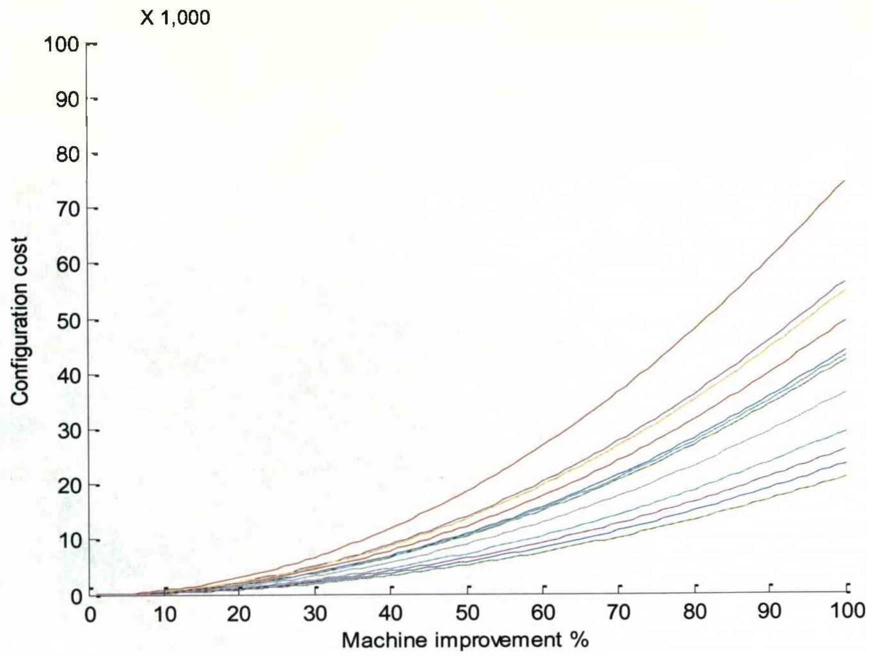


Figure C.22: (Example 2) Random convex non-linear vary cost machine model.

12. Non-linear convex model where bottleneck machines are high cost

$C(x) = kx^2$

Parameter k is randomly generated between 0.005 to 0.0075 for bottleneck machines and 0.002 to 0.003 for non-bottleneck machines.

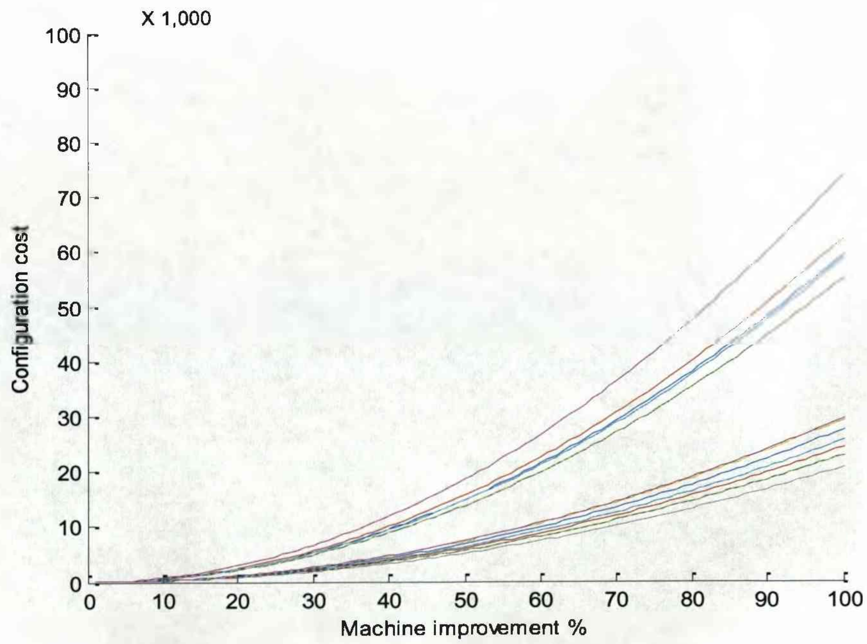


Figure C.23 : (Example 1) Random convex non-linear relationships (Bottleneck machines are high cost).

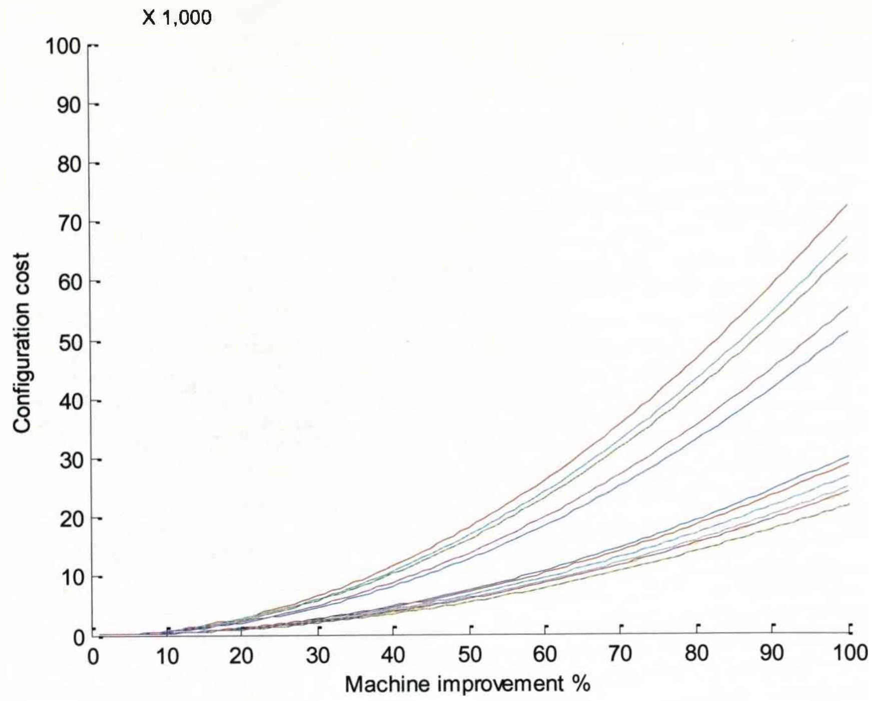


Figure C.24: (Example 2) Random convex non-linear relationships (Bottleneck machines are high cost).

Appendix D
Matlab code for reconfiguration cost models

```
% simple random number generator
% Generating a number in a certain range
% Want 12 of these numbers

clc                % clears the screen
close all        % closes all open figures
clear             % clears the workspace
% read machine improvement data from excel
Imp = xlsread('MD1-improvements','IMP');
Imp=Imp.*100;

% generate random number
disp('Select a Cost Model :');
disp('1. Linear low cost machine');
disp('2. Linear high cost machine');
disp('3. Linear vary cost machine');
disp('4. Non-linear convex low cost machine');
disp('5. Non-linear convex high cost machine');
disp('6. Non-linear convex vary cost machine');
disp('7. Non-linear concave low cost machine');
disp('8. Non-linear concave high cost machine');
disp('9. Non-linear concave vary cost machine');
a=input('Press a number:');

if a==1
    % Linear low cost machine
    for d=1:12
          output(d,1)=rand*0.1+0.2;
    end
end
if a==2
     % Linear high cost machine
     for d=1:12
          output(d,1)=rand*0.25+0.5;
     end
end
if a==3
     % Linear vary cost machine
     for d=1:12
          output(d,1)=rand*0.55+0.2;
     end
end
if a==4
     % Non-linear convex low cost machine
     for d=1:12
           output(d,1)=rand*0.001+0.002;
      end
end
if a==5
     % Non-linear convex high cost machine
     for d=1:12
```

```
            output(d,1)=rand*0.0025+0.005;
        end
end
if a==6
        % Non-linear convex vary cost machine
        for d=1:12
            output(d,1)=rand*0.0055+0.002;
        end
end
if a==7
        % Non-linear concave low cost machine
        for d=1:12
            output(d,1)=rand*1+2;
        end
end
if a==8
        % Non-linear concave high cost machine
        for d=1:12
            output(d,1)=rand*2.5+5;
        end
end
if a==9
        % Non-linear concave vary cost machine
        for d=1:12
            output(d,1)=rand*5.5+2;
        end
end

% Plot the cost model
x=1:1:130; x=x'; n=input('Enter the value of parameter n: ');
for d=1:12
    d1=num2str(d);
    eval(['y' num2str(d1) '(:,d)=output(d,1).*x.^n;'])
    y(:,d)=output(d,1).*x.^n;
end

for d=1:12
k_output(:,d1)=output(1,:);
end


figure(1)
for d=1:12
hold on;
plot(x,output(d).*x);
xlabel('Machine improvement %');
ylabel('Configuration cost');
axis([0 100 0 100])
end

figure(2)
hold on;
%subplot(12,2,d1)
plot(x,y)
eval(['legend(''n=' num2str(n) ''');']);
xlabel('Machine improvement %');
ylabel('Configuration cost');
axis([0 100 0 100])
% Calculate each improvement cost
% use each column from Imp to be a set of x seriers
```

```
a=0.5;

for n=1:6
    for m=1:12
        cost_m(m,n)=output(m).*(Imp(m,n).^a);
        cost=cost_m';
    end
end

str=['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L'];
for d2=1:6
%     Run=strcat('Run_', num2str(d2));
    Run(d2)=(d2);
eval(['figure(' num2str(d2+10) ')'])
hold on
title('Machine Improvement Cost')
for d3=1:12
    if Imp(d3,d2)>0
plot(Imp(d3,d2).*100, cost_m(d3,d2), 'r.')
end
xlabel('Machine Improvement [%]')
ylabel('Normalised Cost []')
if Imp(d3,d2)>0
text(Imp(d3,d2).*100.01, cost_m(d3,d2).*1.01, str(d3), 'Color', 'b')
end
end
axis([0 150 0 10])
end

% output the results to costoutput.xls
 xlswrite('costoutput_md4.xls',Run','cost','A2')
 xlswrite('costoutput_md4.xls', str,'cost','B1')
 xlswrite('CostOutput_md4.xls', cost_m', 'Cost', 'B2')
```

Published with MATLAB® 7.4